Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Lukas R. Bearden

Entitled
An Automated Grid-Based Robotic Alignment System for Pick and Place Applications

For the degree of Master of Science in Mechanical Engineering

Is approved by the final examining committee:

Ali Razban
Co-chair

Tamer Wasfy
Co-chair

Lingxi Li

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Ali Razban

Tamer Wasfy

Approved by: Sohel Anwar 12/04/2013

Head of the Graduate Program Date

AN AUTOMATED GRID-BASED ROBOTIC ALIGNMENT SYSTEM FOR PICK AND PLACE APPLICATIONS

A Thesis
Submitted to the Faculty
of
Purdue University
by
Lukas R. Bearden

In Partial Fulfillment of the
Requirements for the Degree
of
Master of Science in Mechanical Engineering

December 2013
Purdue University
Indianapolis, Indiana

To my Family.
An Oasis of Life in a desert of truth and love.
The only true kind of Family on earth!
4JLT.com

ACKNOWLDEGEMENTS

I would like to thank Dr. Ali Razban for his guidance and patience in the thesis process. He kept me on track and helped me think more deeply and methodically about my work. Furthermore I would like to express my gratitude for the patience he showed in my education, being always eager to help me understand the material he taught and take it deeper.

I would like to thank Dr. Tamer Wasfy for his input in the area of testing and data collection. I would also like to acknowledge the effort he put into educating me through the three courses that I took under his guidance.

I would like to thank Beckman Coulter for allowing me to use its resources to fund my thesis experimentation. I am also grateful for the use of equipment, especially a robot to make my system a reality. Specifically I would like to acknowledge Wayne Schmidt for his support of the project. I would like to thank Tim Sherrill for his encouragement and facilitation of the project. The support he showed frequently spurred me on to continue with my research

I would like to thank Sam Rizzotte for his encouragement, as my manager, to continue my research and his willingness to allow a flexible work schedule. I would like to acknowledge the valuable insight and input that Joseph Shriner contributed in developing my control system and operating the robot. Furthermore I would like to express my gratitude toward C. Stephen Kraihanzel for his input, encouragement, and inspiration.

I would like to specifically thank John Fitzpatrick for his effort in helping me develop the electronic components of my setup. His many hours and patient guidance are deeply appreciated.

I would like to acknowledge the encouragement and friendship received from Garrett Yesmunt and Nathan Fitzpatrick throughout my thesis research, as well as throughout my educational career.

I would also like to acknowledge all those who have supported me during the research and writing of my thesis. The encouragement continually kept me going and without which I would not have finished this thesis. The list of people is too long to record, but my gratitude is none-the-less immense.

Finally I would like to thank “MH” for the continued support, encouragement, and help throughout the course of my thesis.

TABLE OF CONTENTS

Page

APPENDICES

LIST OF TABLES

Appendix Table

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Appendix Figure

LIST OF DEFINITIONS

Titer Plate: a liquid handling plate used in diagnostic instruments for running various chemistry and biological tests. [Appendix E-1]

Laser Diode: small chip-sized laser made from semiconductor material that has a high output for its size, over 100 watts at times [1].

Photodiode: "Silicon photodiodes are semiconductor devices responsive to high-energy particles and photons. Photodiodes operate by absorption of photons or charged particles and generate a flow of current in an external circuit, proportional to the incident power." [2]

Home Position: (Target Location) position the robot is aligned to in the upper left side of the Grid Array

Collimate: make (rays of light or particles) accurately parallel [3]

Concatenate: link (things) together in a chain or series [4], especially data inputs of type String to create a single input

ABSTRACT

Bearden, Lukas R. M.S.M.E., Purdue University, December 2013. An Automated Grid-Based Robotic Alignment System for Pick and Place Applications. Major Professors: Ali Razban and Tamer Wasfy.

This thesis proposes an automated grid-based alignment system utilizing lasers and an array of light-detecting photodiodes. The intent is to create an inexpensive and scalable alignment system for pick-and-place robotic systems. The system utilizes the transformation matrix, geometry, and trigonometry to determine the movements to align the robot with a grid-based array of photodiodes.

The alignment system consists of a sending unit utilizing lasers, a receiving module consisting of photodiodes, a data acquisition unit, a computer-based control system, and the robot being aligned. The control system computes the robot movements needed to position the lasers based on the laser positions detected by the photodiodes. A transformation matrix converts movements from the coordinate system of the grid formed by the photodiodes to the coordinate system of the robot. The photodiode grid can detect a single laser spot and move it to any part of the grid, or it can detect up to four laser spots and use their relative positions to determine rotational misalignment of the robot.

Testing the alignment consists of detecting the position of a single laser at individual points in a distinct pattern on the grid array of photodiodes, and running the entire alignment process multiple times starting with different misalignment cases. The first test provides a measure of the position detection accuracy of the system, while the second test demonstrates the alignment accuracy and repeatability of the system.

The system detects the position of a single laser or multiple lasers by using a method similar to a center-of-gravity calculation. The intensity of each photodiode is multiplied by the X-position of that photodiode. The summed result from each photodiode intensity and position product is divided by the summed value of all of the photodiode intensities to get the X-position of the laser. The same thing is done with the Y-values to get the Y-position of the laser. Results show that with this method the system can read a single laser position value with a resolution of 0.1mm, and with a maximum X-error of 2.9mm and Y-error of 2.0mm. It takes approximately 1.5 seconds to process the reading.

The alignment procedure calculates the initial misalignment between the robot and the grid of photodiodes by moving the robot to two distinct points along the robot's X-axis so that only one laser is over the grid. Using these two detected points, a movement trajectory is generated to move that laser to the X = 0, Y = 0 position on the grid. In the process, this moves the other three lasers over the grid, allowing the system to detect the positions of four lasers and uses the positions to determine the rotational and translational offset needed to align the lasers to the grid of photodiodes. This step is run in a feedback loop to update the adjustment until it is within a permissible error value. The desired result for the complete alignment is a robot manipulator positioning within ±0.5mm along the X and Y-axes. The system shows a maximum error of 0.2mm in the X-direction and 0.5mm in the Y-direction with a run-time of approximately 4 to 5 minutes per alignment. If the permissible error value of the final alignment is tripled the alignment time goes down to 1 to 1.5 minutes and the maximum error goes up to 1.4mm in both the X and Y-directions. The run time of the alignment decreases because the system runs fewer alignment iterations.

1. INTRODUCTION

1.1 Need

As the uses and applications of automation expand, so also does the need to accurately align robotics and equipment to perform their tasks. To perform this alignment, it is essential to have an inexpensive, efficient, and robust alignment system. Many sensory systems currently exist such as vision, optical, and contact systems. Each system has an advantage depending on the application and it is essential to apply the best alignment method to the appropriate application. The importance of having a system that works rapidly is specifically seen in areas where frequent alignment is necessary. The proposed grid-based sensor alignment system promises to be inexpensive, scalable, and computationally straightforward.

This system is specifically designed to serve in the alignment of automation systems that move *titer* liquid handling plates from one processing point to another. An example system developed by Beckman Coulter, Inc. is shown in Figure 1.1.

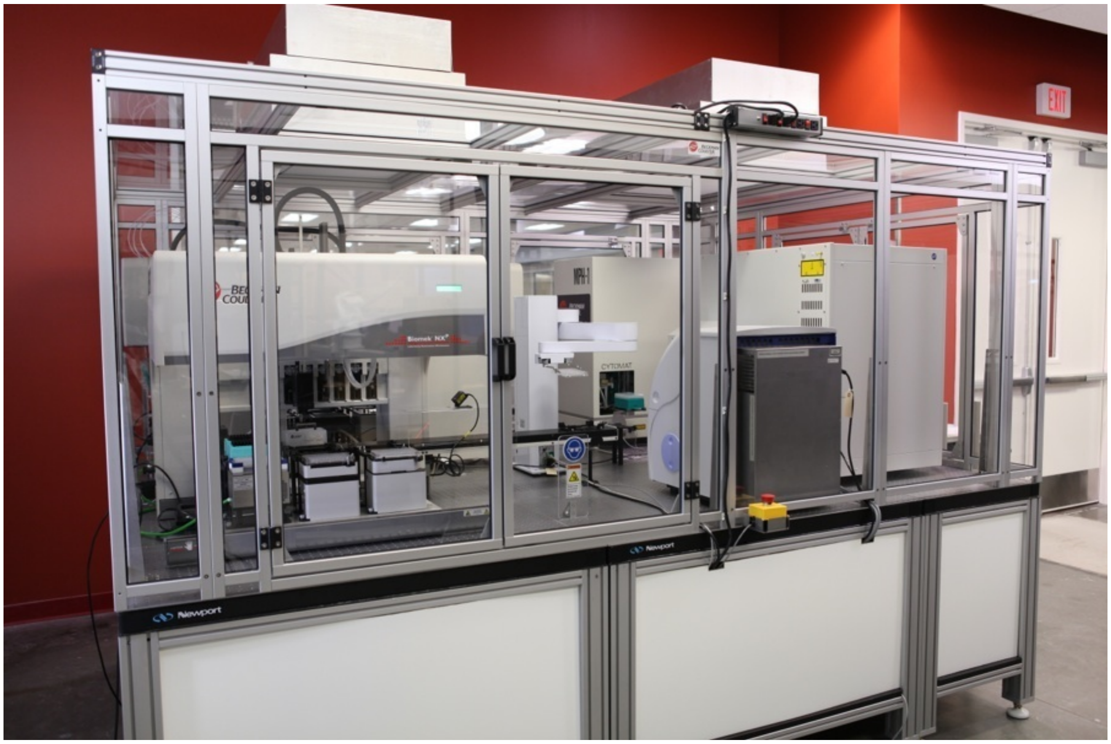


Figure 1.1 Integrated Analysis Automation System, Beckman Coulter, Life Science Division Headquarters, Indianapolis [Appendix E-2]

The current method of alignment is to use a teach pendent to visually align a *titer plate* to a location like that shown in Figure 1.2. This is subject to inaccuracies of visual alignment and becomes difficult to do in the center of enclosed systems like that in Figure 1.1. Another commonly used method is to turn off the joint motors of the robot, but leave the encoders active. An operator then moves the robot into place and instructs the system to remember the position. Because the encoders are still active, the robot controller records the position, allowing the robot to repeatedly return to this point.

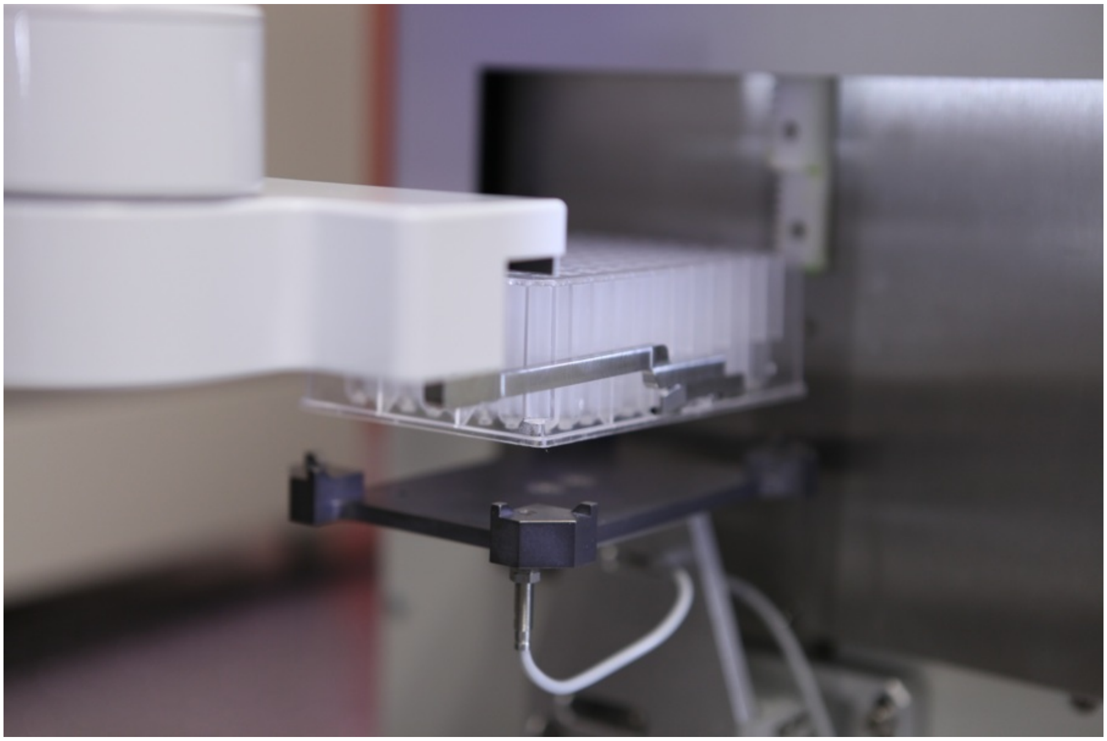


Figure 1.2 Titer Plate Being Aligned to Processing Position [Appendix E-3]

1.2 Literature Review

Likely the most common alignment method used recently is the vision system using a light source and a camera that detects specified markers, objects, or geometries within its view and processes the information in the form of images. Much research is going into algorithm development and improvement for such alignment systems, but a primary disadvantage of this solution compared to the proposed solution described here is the complexity of distinguishing specific objects or reference points based on varying light intensities in an image. Chapter twelve of Robot Modeling and Control [5] introduces this point, and then it is further understood through investigation of the conference paper, "Pose Alignment Of An Eye-In-Hand System Using Image Morphing" [6] which requires the system to "learn" the a shape in one orientation to detect a similar object in another orientation. To move into position to grasp an object, it must apply rotational and translational transformations to the image until the image matches the object of interest. Additionally, Horaud, et al. described a method of detecting and grasping an image in one environment based on images used from grasping the object in test environments. Essentially, the system described in "Visually Guided Object Grasping," is a machine learning vision system [7].

Although visions systems can be used in a wide variety of applications, the algorithms used in object detection and path planning rely on many parameters, making them complex. They require appropriate lighting, and often, prior knowledge of what to detect. Therefore, they do not pose a simple solution for continually changing alignment tasks. Vision systems are more ideal for complex alignment routines that would require multiple less-complicated methods to achieve the same alignment.

In 2003 Beckhart et al. patented an alignment concept involving placing a light emitting and detecting sensor on a robotic manipulator for use in a clean room facility. Rough alignment sensors are placed along the gross movement path of the manipulator. As the manipulator approaches the final destination, a fine-alignment sensor is used to make the precise alignments. In this case, the system is used to align a robot moving silicon wafers. The alignment is achieved by using a light beam that strikes a reflective patch and reflects back to the receiver-sensor combination. In a non-environmentally controlled area, contamination of the reflective surfaces would change the sensor reading, therefore liming the applications of this method. For the application that was patented, that was not a concern because it was essential that the environment stay dust-free [8].

More recently, the company HighRes Biosolutions has developed a device called the MicroTeach^{TM} used for aligning *titer plates*. This device utilizes a sensor array held by the manipulator. A fixed plate is placed in the destination location and the robot moves the sensor array until it touches each of the inside edges of the fixed plate, determining the spatial position and yaw of the plate. It then touches the bottom surface of the sensor array to the bottom of the fixed plate to determine the pitch and roll of the system. The downside of this system is that the sensor array used is approximately 50 to

60mm tall and will not fit into the space needed for certain applications See Figure 1.3 for pictures of the system [9].

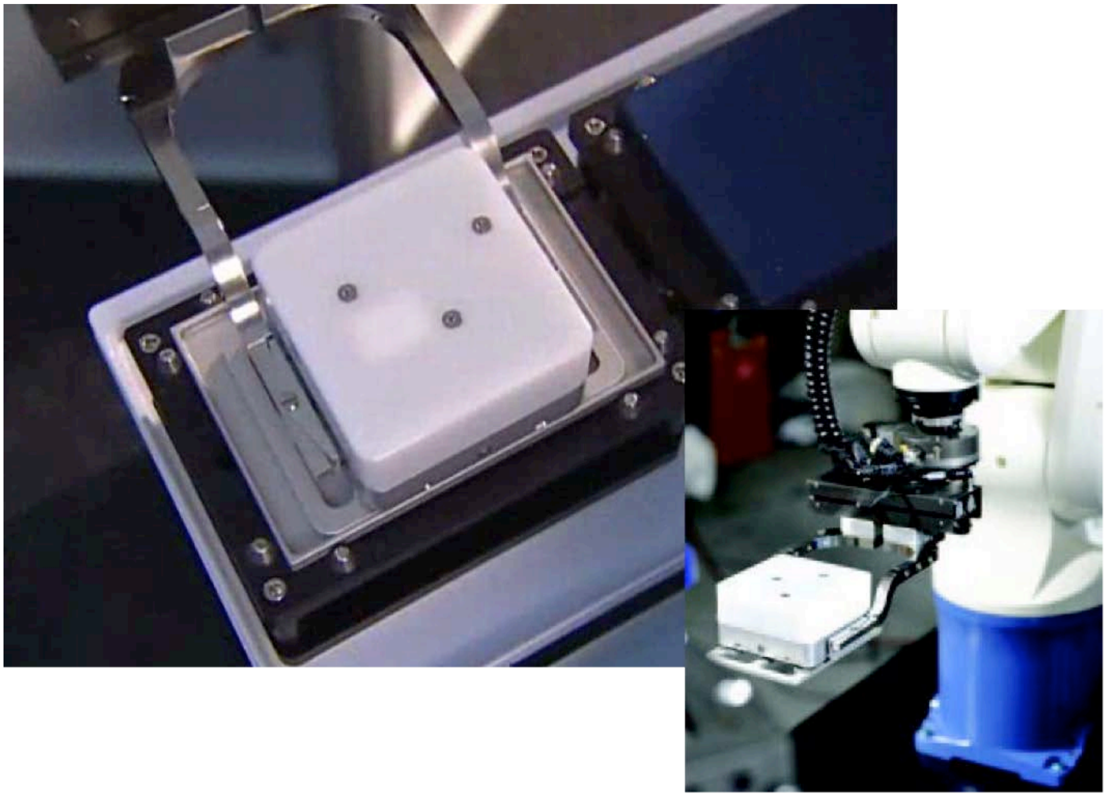


Figure 1.3 HighRes Biosolutions MicroTeach™ [9]

The paper, “A Novel Multi-DOF Precision Positioning Methodology Using Two-Axis Hall-Effect Sensors” by Kawato and Kim describes a plate alignment method using two magnetic Grid Arrays and two to four Hall-effect sensors. Each magnetic array produces a magnetic field along each of the X and Y-axes. The results show promise with the ability to achieve micrometer accuracy, but the physical setup of the system is quite large, being nearly 330mm long on each side. The fact that a dual-axis magnetic array is necessary to align the system makes this alignment method difficult to use in compact applications. Furthermore, the intent of the system is to precisely align surfaces such as mirrors that benefit from continual alignment adjustment versus a one-time alignment [10].

DUMA Optronics Ltd. uses a light-detecting sensor that allows the position of the laser striking the sensor to be determined with a high resolution within the sensing region. Although a high resolution is possible, the size of each sensor (77 to 100 mm^2) reduces the possible placement density for grid applications. Additionally, because the sensors are round the area lost in the corner between four sensors would cause a large dead-space that would not detect the laser location. The circular sensor area is broken into four quadrants separated by a small gap. A connected computer compares the signal output from each quadrant to determine the location of the laser. Therefore, to use multiple light-detecting sensors would require performing calculations for each individual sensor and for the combined positioning using them. The purpose of the sensor is to very accurately detect a laser spot in a small area, meaning it is not ideal for use with other sensors to detect a laser spot somewhere in a larger area [11].

In his paper "The Frame of Controlled Active Vision", N.P. Papanikolopoulos described a system that utilized a camera mounted to the robot manipulator. The system includes a rapid-response controller achieving a response frequency up to 30Hz to update the robot location to allow for continual adjustment while the manipulator approaches a moving object. Furthermore, to allow the system to detect the appropriate object, it is necessary that parameters of the basic shape and motion of the object are fed into the system prior to instructing it to find the object. For these two reasons this system is complex, requiring a sophisticated controller to continually map out a motion path for the robot. Furthermore, it must decipher an image based on prior information of the object of interest to determine the location of the object [12].

Another sensing method is using contact force detection. This requires that the object of interest come in contact with sensors to detect its location. Various methods and applications are described by Jacobsen et al. in *Tactile Sensing System Design Issues in Machine Manipulation* [13] and by Ohara et al. in *Robot Operation Based on Model Updated by Sensory Data* [14].

Both discuss the use of tactile sensing in manipulation to adjust the manipulator grasp force and position based on the output of the tactile sensors as they come in contact with an object. This is useful in making small alignment adjustments, but is not ideal for displacements larger than a few millimeters such as desired in the application covered by this thesis.

For some SCARA alignment processes, the position to pick or place the robot is taught be using one of two methods, or a combination of these methods. The first is to move the robot by hand in a "lose" teach mode that disables the motors but continues to track the encoder positions. This allows the operator to move the robot to a teach point, activate the gripper to grasp a titer plate, and then teach the position to the robot. The second method is to use a teach pendent, either physical or on a computer, to move the robot into a position and then teach the position. Both of these methods are described in the "Setup Guide & User's Manual Revision 2.2 PreciseFlex 400 SCARA on SAMI EX System" paper written by Beckman Coulter [15].

1.3 Proposed System Overview

To mitigate high-costs and complex system teaching, a grid-based alignment system is implemented in this thesis. It consists of *laser diodes* held by a robot gripper above a grid array of *photodiodes*. Based on the detected laser positions, the system calculates the movements required to align the lasers to the Grid Array.

A robot holds four *laser diodes* above an array of *photodiodes*. The system first uses a single laser along the edge of the grid to determine the difference between the robot coordinate frame and the grid coordinate frame. Based on the relation, the system moves the primary laser to the origin point of the array of photodiodes. The system then finds the position of all four lasers and uses this to complete the alignment of the system by translating and rotating the robot gripper to align the lasers to the array of photodiodes. This last functionality is performed within a feedback loop that repeats the position detection and movement steps until the alignment of the lasers is within an acceptable error tolerance of the desired location.

The following chapters describe the alignment system and methods by going through the functionality of each component and the calculations required for each algorithm, and then describes the tests used to evaluate the functionality and accuracy of the system and their results. Chapter 2 gives an overview of the components of the system and how they interact. Chapter 3 discusses details of how the system determines the location of each laser and how the system calculates the system moves based on the laser locations. An in-depth description of each system component is laid out in Chapter 4, with the computer control system thoroughly described in Chapter 5. Chapter 6 discusses two steps required for setting up the system, finding the center of the laser plate and setting a threshold value for the photodiode laser detection. The results are then laid out and analyzed in Chapter 7. Finally Chapter 8 concludes the thesis and discusses possible future research and system improvements.

2. ALIGNMENT SYSTEM DESCRIPTION AND HIGH LEVEL FUNCTIONALITY

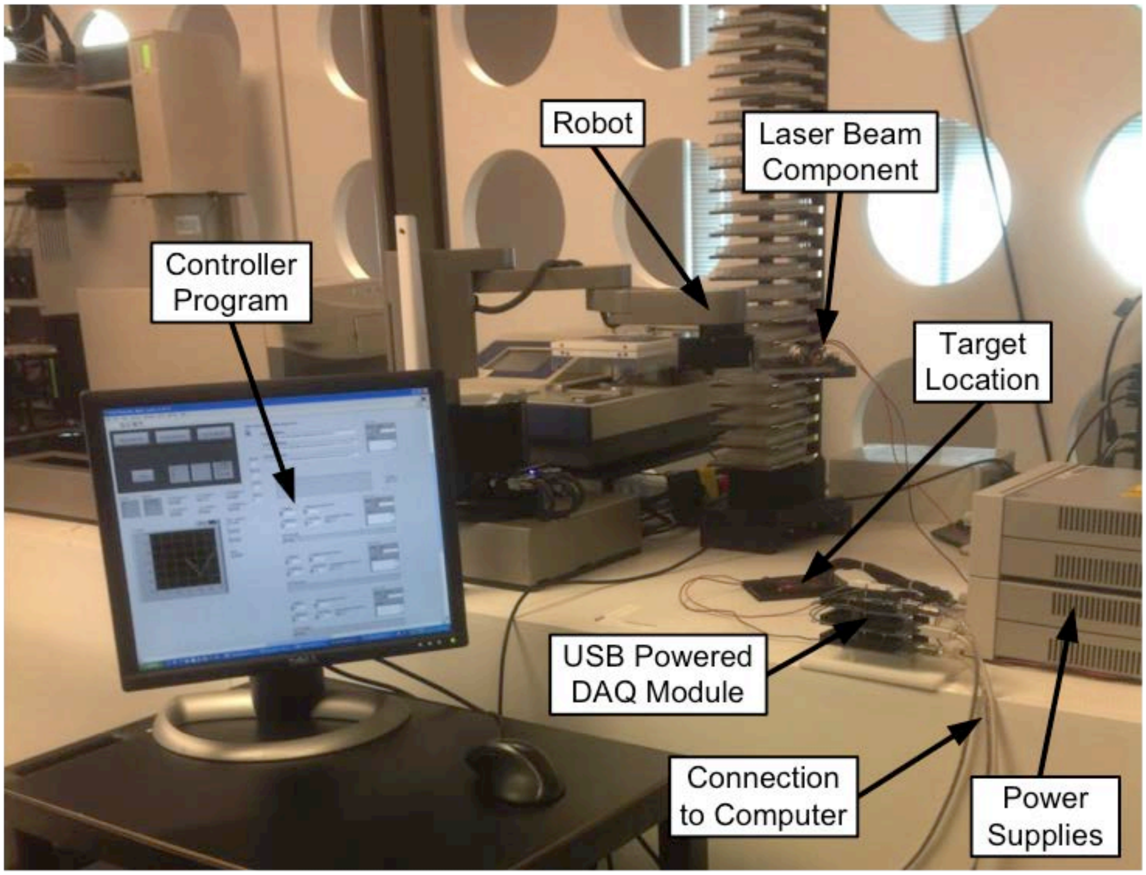


Figure 2.1 System Introduction

The system consists of four main components, a signal sending module, a signal-receiving module, a controller, and a robot. The signal-sending module contains 4 laser diodes and is held in the robot gripper. The signal-receiving module is composed of a grid of photodiodes and sits in the aligned position, also known as the destination position. The controller consists of a data acquisition unit (DAQ), a computer, and a software controller. The robot is a SCARA robot used to move the signal-sending module into place, aligning it to a "teach" position for future pick or place operations in that location. The system is laid out in Figure 2.1 above.

The sending module is the **Laser Beam Component** gripped by the **Robot**. Below the Laser Beam Component is the receiving module labeled **Target Location**. Both the sending module and receiving module obtain power from the **Power Supplies**, and the receiving module transmits data through wires to the **USB Powered DAQ Module**. This converts the analog inputs of the receiving module into three digital signals that are sent through the three cables labeled **Connection to Computer**.

The **Controller Program** resides on the computer. It takes in the signals from the DAQ and uses them to calculate translations and rotations to move the robot gripper into place above the receiving module. Once the feedback alignment procedure has sufficiently aligned the two plates, the program is complete.

The sending module grasped by the robotic gripper houses four small laser diodes controlled by a single control module shown in Figure 2.2. It gets its power from a power supply, and is not connected to the computer controller.

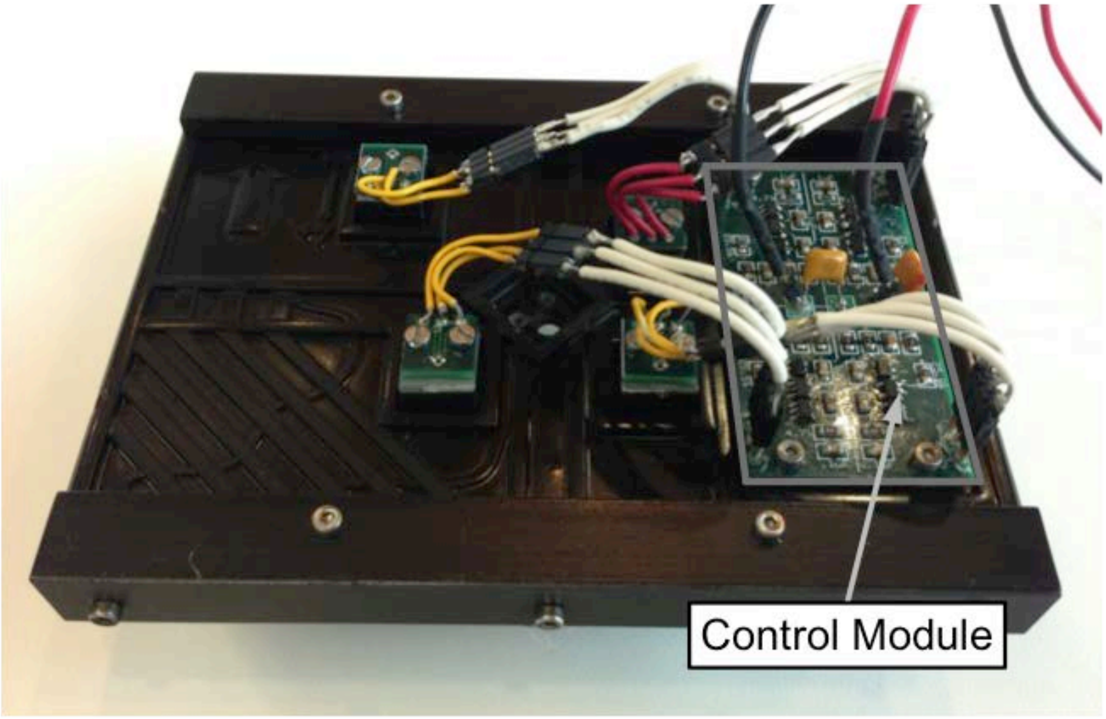


Figure 2.2 Laser Control Module

The receiving module consists of a six by six Grid Array of photodiodes as shown in Figure 2.3. The photodiodes receive power from a power supply, and connect to a data acquisition unit that translates the analog output of each photodiode into a single combined digital signal that the computer can process.

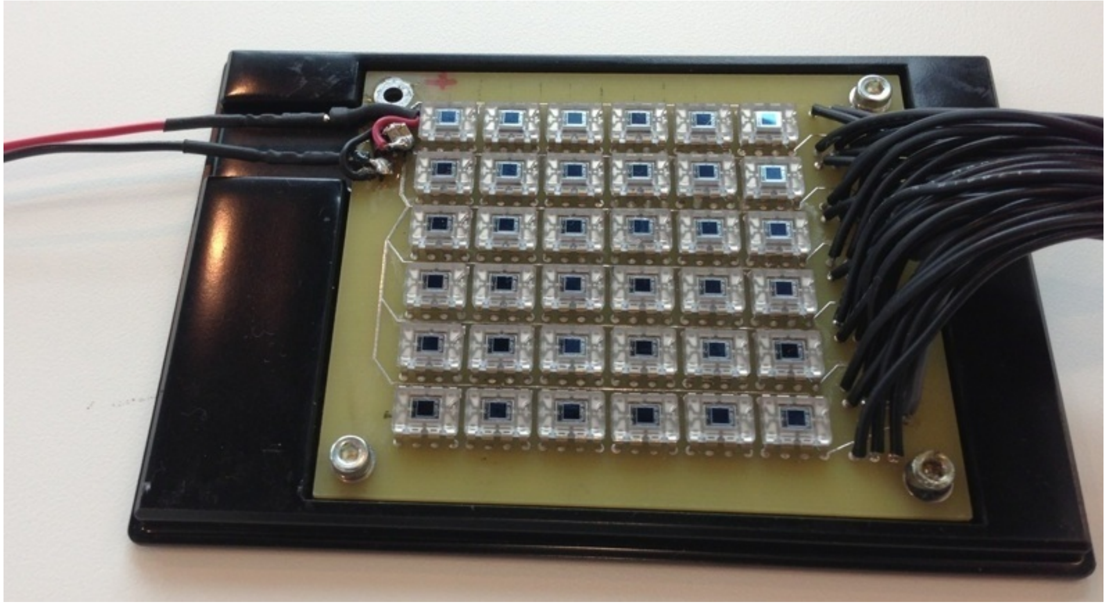


Figure 2.3 Photodiode Grid Array

The data acquisition unit consists of a USB powered module that receives an analog signal from each photodiode and outputs a digital signal to the computer. Figure 2.4 shows the series of three boards that forms the DAQ unit.

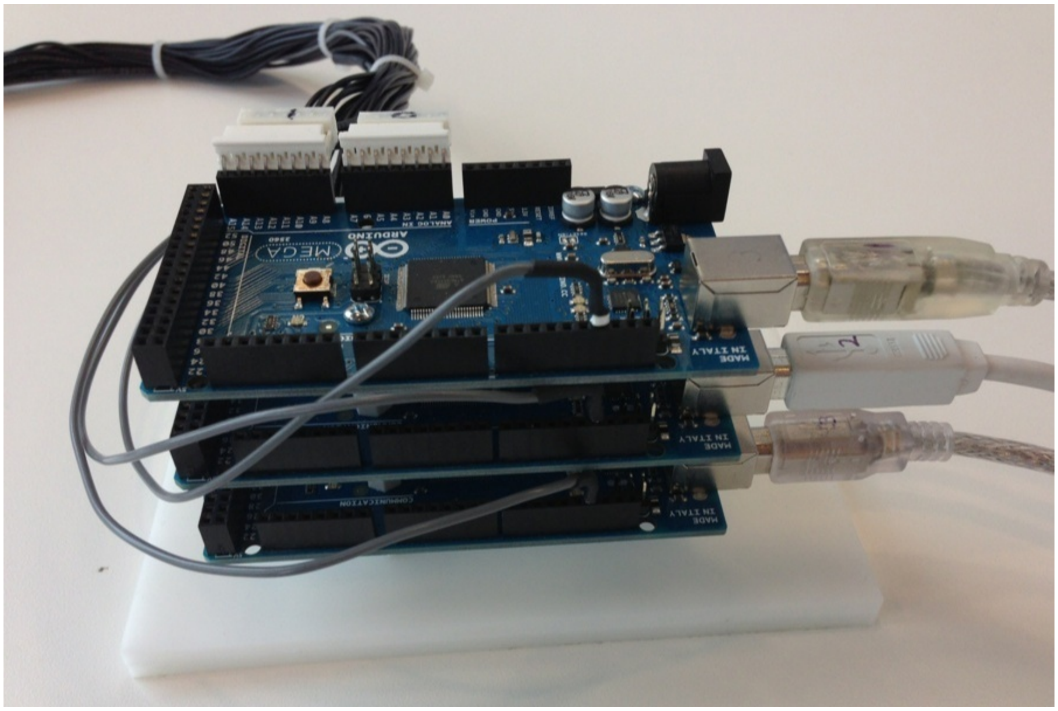


Figure 2.4 Data Acquisition Unit

The computer then processes the data using a combination of three LabVIEW programs. Two programs take in the output of the photodiodes and calculate the position of the laser(s) on the grid using the output signal of each photodiode. One of these programs analyzes the grid as a whole and is capable of detecting only a single laser diode. The other program discretizes the grid into four quadrants, allowing the system to detect up to four laser diodes at once.

Once the position detection programs have run, the position of the single laser in the full grid is given to a third program, which is the main control algorithm to create motion commands to send to the robot. In another part of the motion control, the position of each of the four lasers is detected and passed off to the controller. This information is used to detect the rotations needed to align the gripper to the target location. The components of the system are shown in Figure 2.5 below. The sending module is **Module A**. The receiving module is **Module B**, and the data acquisition unit and the computer with the LabVIEW software are represented by the **Sensor Controller** component shown. **Module B** sits in the **Target Location**. This is the location that is taught to the overarching automation system (that the **SCARA Robot** is a component of) after the alignment process, allowing the robot to return to this position later.

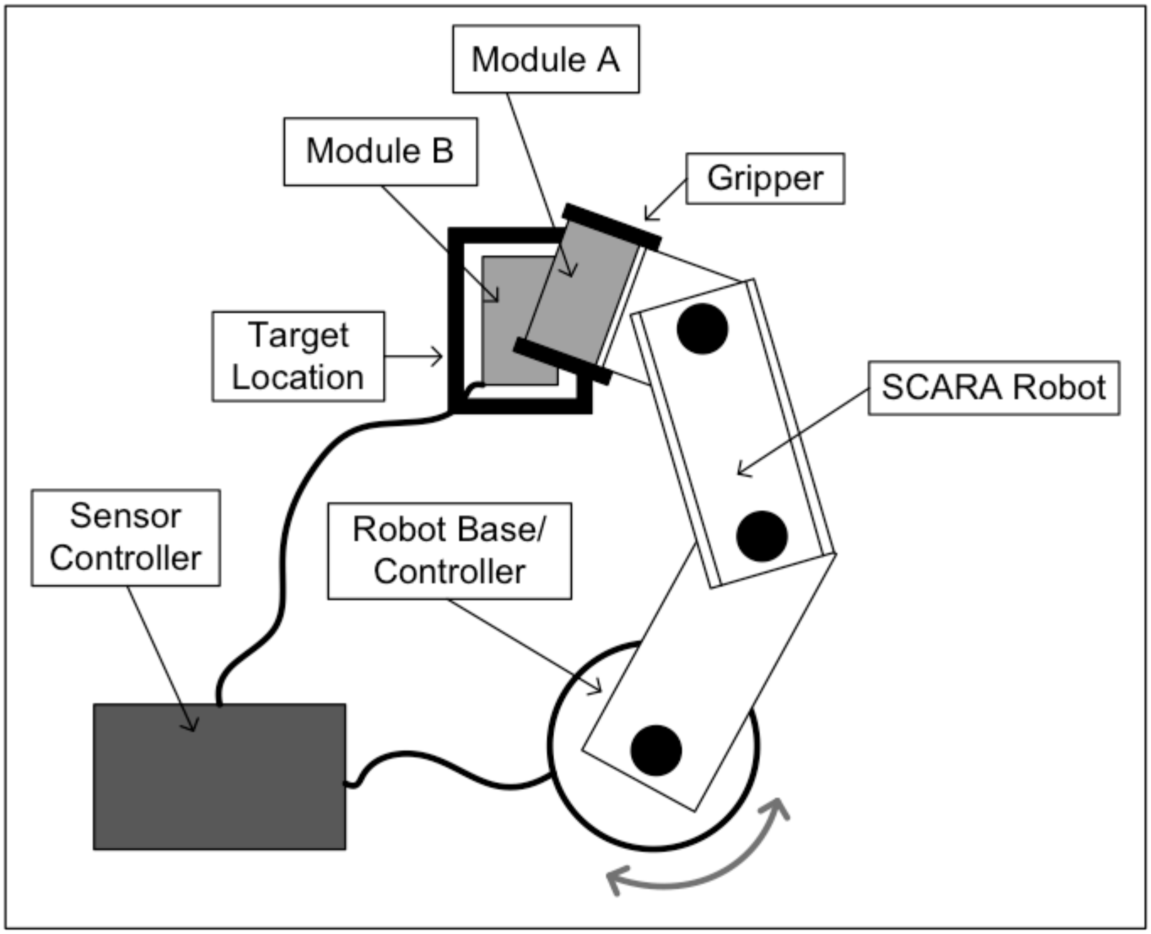


Figure 2.5 Alignment System

A high-level view of the system is shown in the following block diagram in Figure 2.6. It shows that the computer control system takes in the analog signals from the photodiodes as an input and provides outputs as robot commands and an on-screen display to track progress and show results.

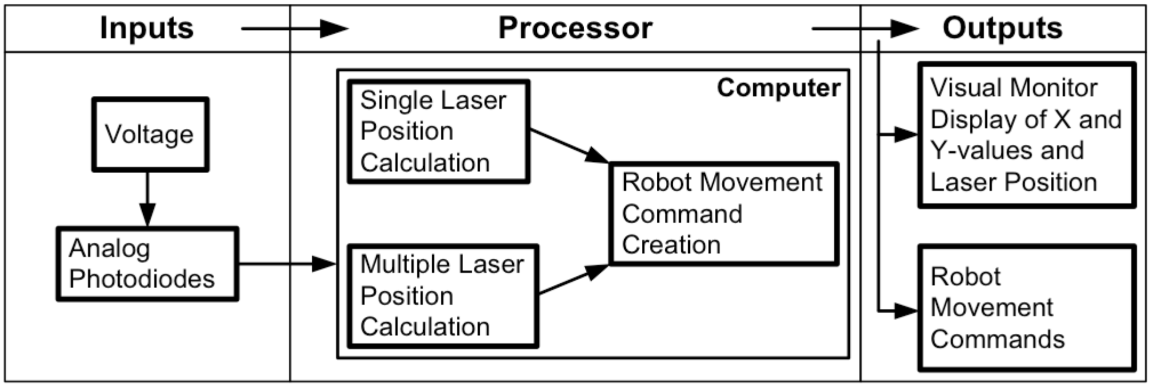

Inputs
Processor
Outputs
Voltage
Analog Photodiodes
Computer
Single Laser Position Calculation
Multiple Laser Position Calculation
Robot Movement Command Creation
Visual Monitor Display of X and Y-values and Laser Position
Robot Movement Commands

Figure 2.6 Alignment System Overview

3. METHOD OVERVIEW

The system starts by detecting the position of an individual laser, and then detects the position of multiple lasers. Using the detected positions, it calculates the alignment of the system and the necessary movements to adjust the alignment. The two ways that the system views the grid are shown in the first portion of this section. The rest of the section discusses the method by which the positions detected by the grid are used to calculate movements to the *Home Position*, followed by the rotations and translations needed to align all four lasers.

The grid-based system uses geometry, trigonometry, and a transformation matrix to determine the transformations required to move the robot from one orientation to another. The advantage of the grid lies in the intuitive Cartesian reference frame, despite the fact that the sign of the Y-axis is reversed. Because of the grid nature, a scaling of the grid requires no change in the alignment concept.

The full grid is represented in Figure 3.1 below. This is used when detecting a single laser, but to detect up to four lasers at once, it is necessary for the system to discretize the grid into quadrants as shown in Figure 3.2.

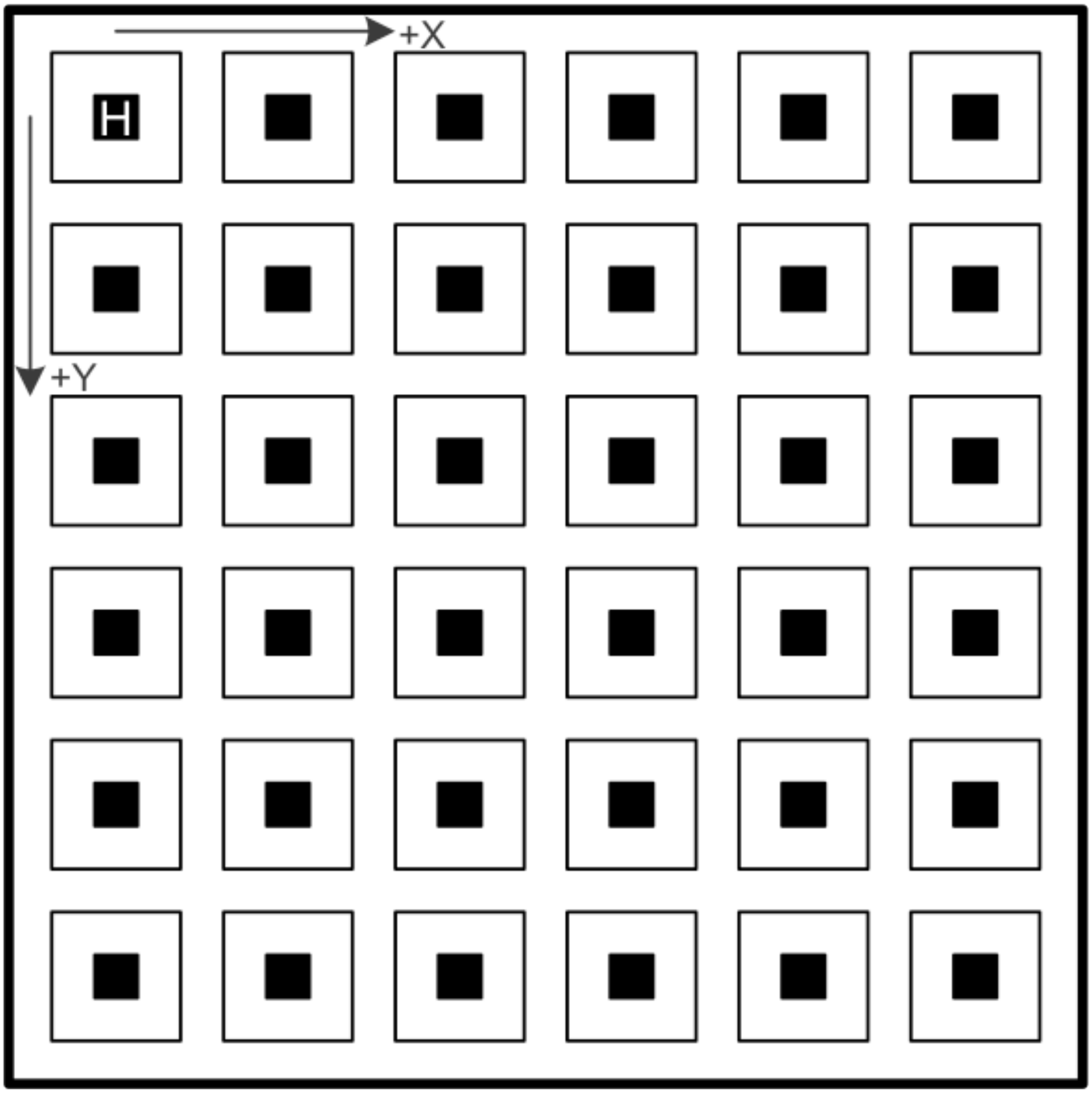


Figure 3.1 Full Grid Array for Single-Laser Detection

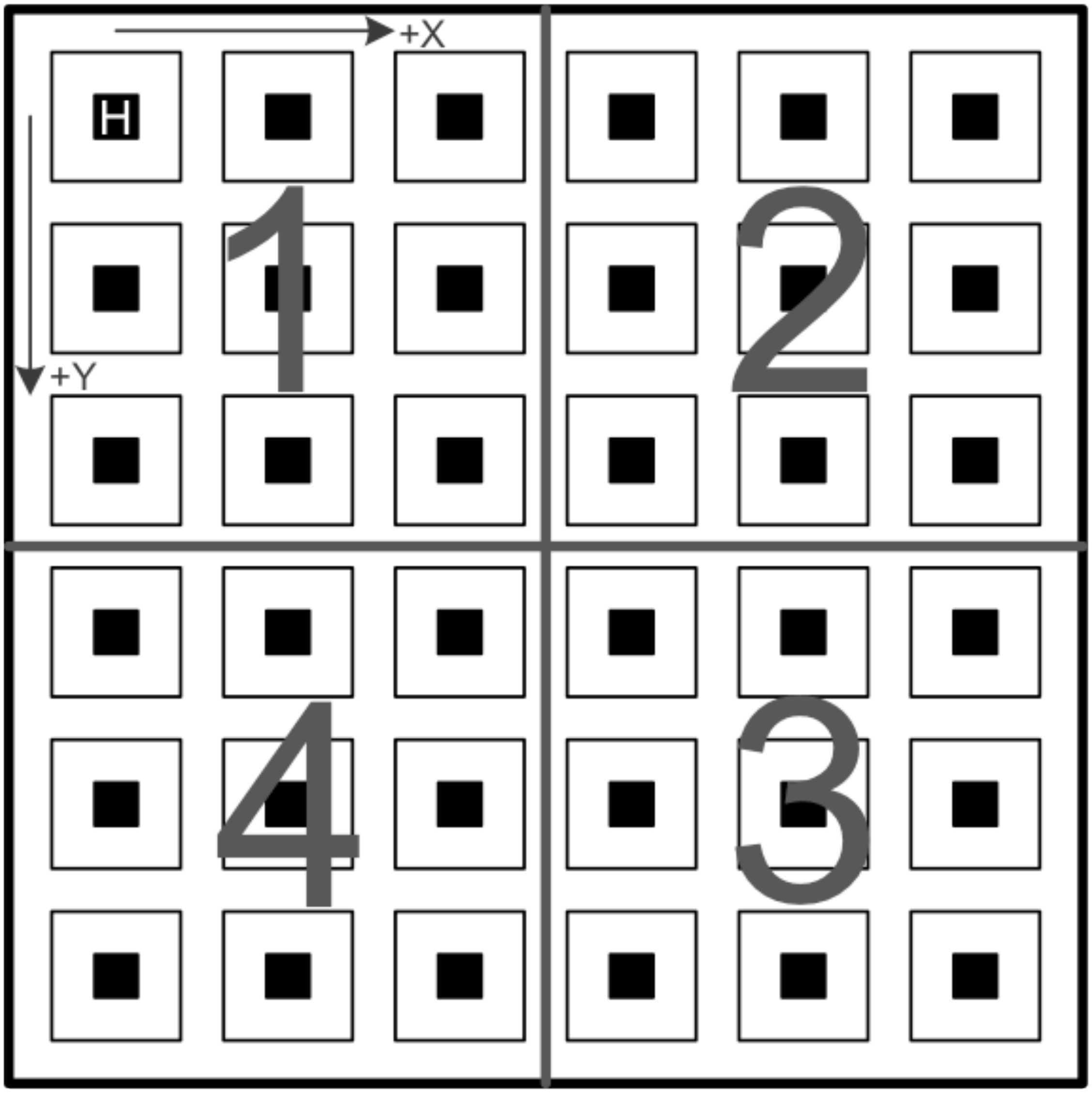


Figure 3.2 Quadrant Discretized Grid Array for Multi-Laser Detection

3.1 Alignment Step 1

For the first step in the alignment when the lead laser is aligned to the origin point, the X and Y-distances are output in the grid reference frame. These are then transformed into the robot reference frame using the method shown below in equations (3.1) - (3.3), which come from the rotation matrix shown by equation (3.4).

$$\alpha = \tan^{-1}\left(\frac{Y2-Y1}{X2-X1}\right) \tag{3.1}$$

$$X_{robot} = X2\cos(-\alpha) - Y2\sin(-\alpha) \tag{3.2}$$

$$Y_{robot} = -X2\sin(-\alpha) - Y2\cos(-\alpha) \tag{3.3}$$

$$\begin{Bmatrix} X_{robot} \\ Y_{robot} \end{Bmatrix} = \begin{bmatrix} \cos(-\alpha) & -\sin(-\alpha) \\ -\sin(-\alpha) & -\cos(-\alpha) \end{bmatrix} \begin{Bmatrix} X2 \\ Y2 \end{Bmatrix} \tag{3.4}$$

Where **α** is the angle between the X-axis of the robot frame and the X-axis of the grid frame, **(X2, Y2)** is the location of the lead laser before moving to the Home Position of the grid, **(X1, Y1)** is the location of another point along the robot X-axis used in calculating the robot X-vector, and $\mathbf{X_{robot}}$ and $\mathbf{Y_{robot}}$ are the distances that the robot moves in the robot coordinate frame to get to the Home Position.

Figure 3.3 provides a pictorial representation of the conversion between the grid reference frame and the robot reference frame.

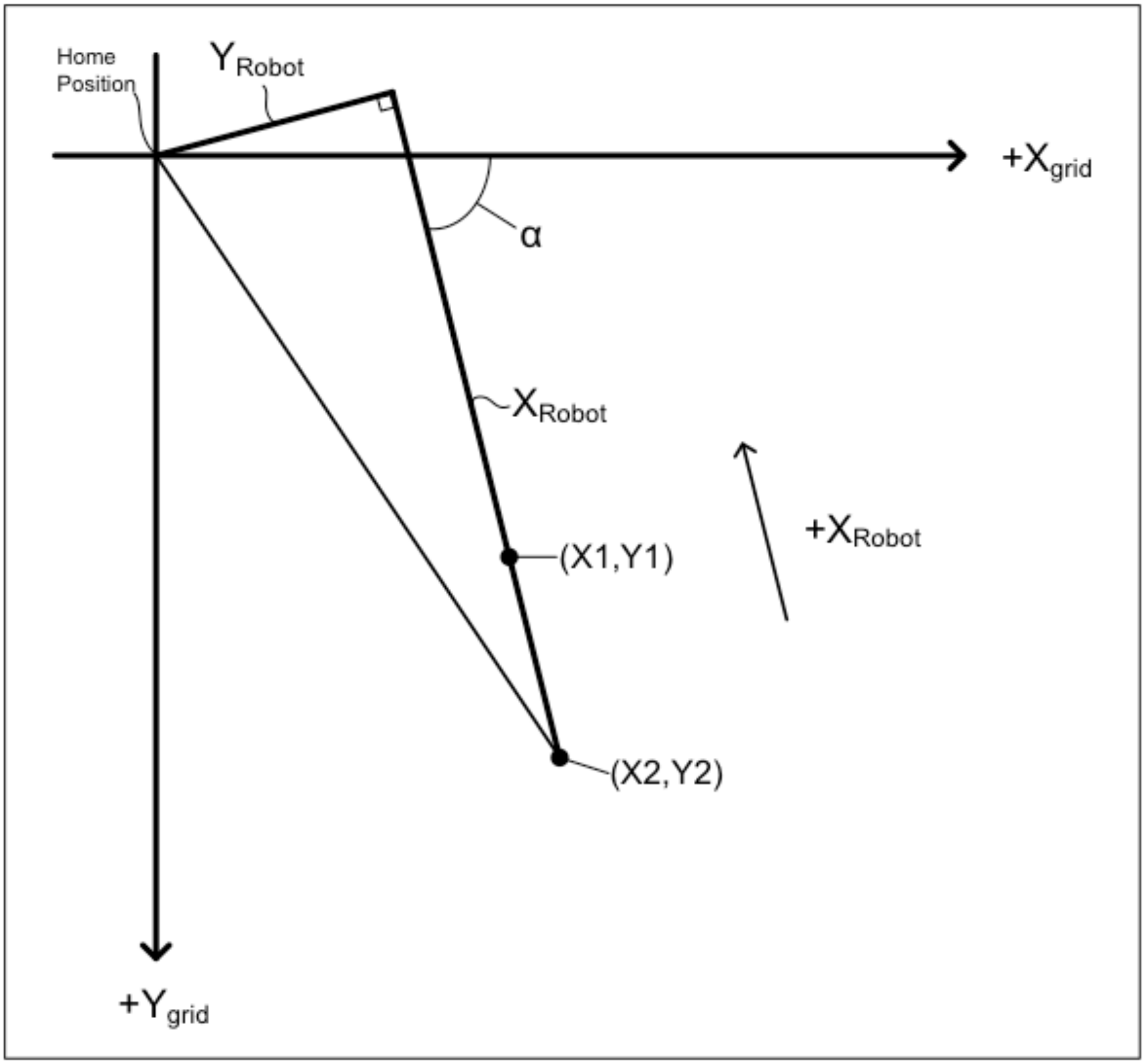


Figure 3.3 Conversion Between Grid and Robot Reference Frames

The points **(X1, Y1)** and **(X2, Y2)** shown in Figure 3.3 above are used to calculate the vector of the positive X-axis of the robot. Based on this vector, the angle **α** between the +X-axis of the grid and the +X-axis of the robot is determined. After finding the angle, equations (3.2) and (3.3) above are used to calculate $\mathbf{X_{robot}}$ and $\mathbf{Y_{robot}}$ as shown in the figure.

3.2 Alignment Step 2

The second step of the alignment requires a conversion between the grid reference frame and the tool reference frame. Additionally, a rotation is calculated to align the lasers of Module A with the photodiodes of Module B. To develop the relationship between the coordinate frames, the angle between the vector formed by the center of Module B and the lead laser and the vector formed by the origin photodiode and the center of the Grid Array is calculated. Based on this angle, the equations (3.5) - (3.7) are used to rotate and move the robot into place.

$$L = 2\sqrt{25^2 + 25^2}\sin\left(\frac{\theta}{2}\right) \tag{3.5}$$

$$X = L\sin\left(135° - \frac{(180° - \theta)}{2}\right) \tag{3.6}$$

$$Y = L\cos\left(135° - \frac{(180° - \theta)}{2}\right) \tag{3.7}$$

Where X and Y are the translations required in the grid coordinate frame to align the center of the laser plate to the center of the grid plate. The equations are developed based on the geometry shown below in Figure 3.4.

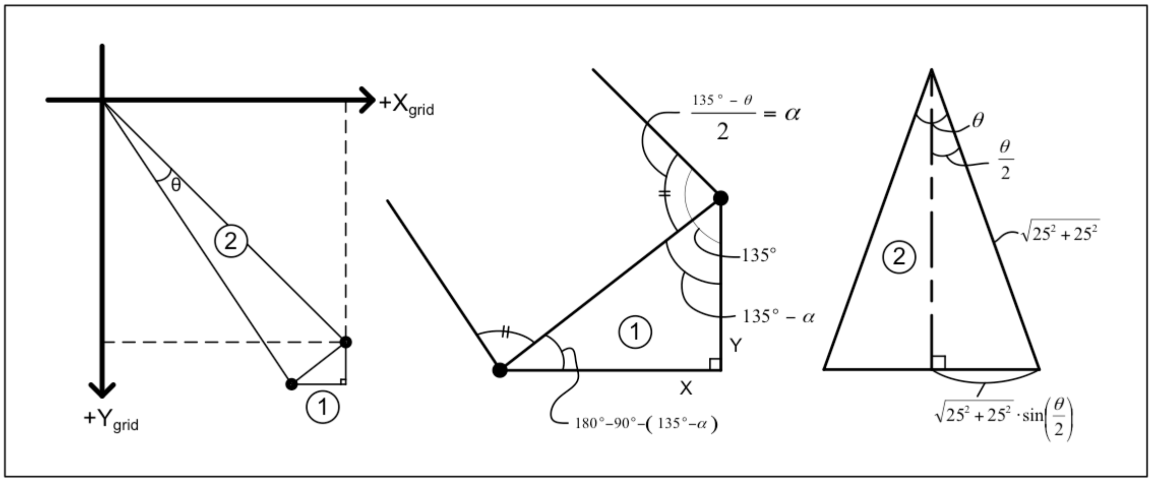

$+X_{grid}$
$+Y_{grid}$
θ
2
1
$\frac{135° - \theta}{2} = \alpha$
135°
$135° - \alpha$
Y
X
$180° - 90° - (135° - \alpha)$
θ
$\frac{\theta}{2}$
$\sqrt{25^2 + 25^2}$
$\sqrt{25^2 + 25^2} \cdot \sin\left(\frac{\theta}{2}\right)$

Figure 3.4 Second Step Alignment Geometry - Method 1

After testing, it was found that this method does not work reliably due to error in the system position detection, as discussed later, although it is more versatile because it is valid as long as at least two lasers are on the Grid Array. Therefore, another method, shown briefly below, is used and discussed further in Chapter 5. The intersections of the equations of the diagonal line between points one and three, and between points two and four is calculated using equations (3.8) - (3.31). Figure 3.5 shows the equation of each diagonal and labels the calculated center position **(Dx, Dy)**.

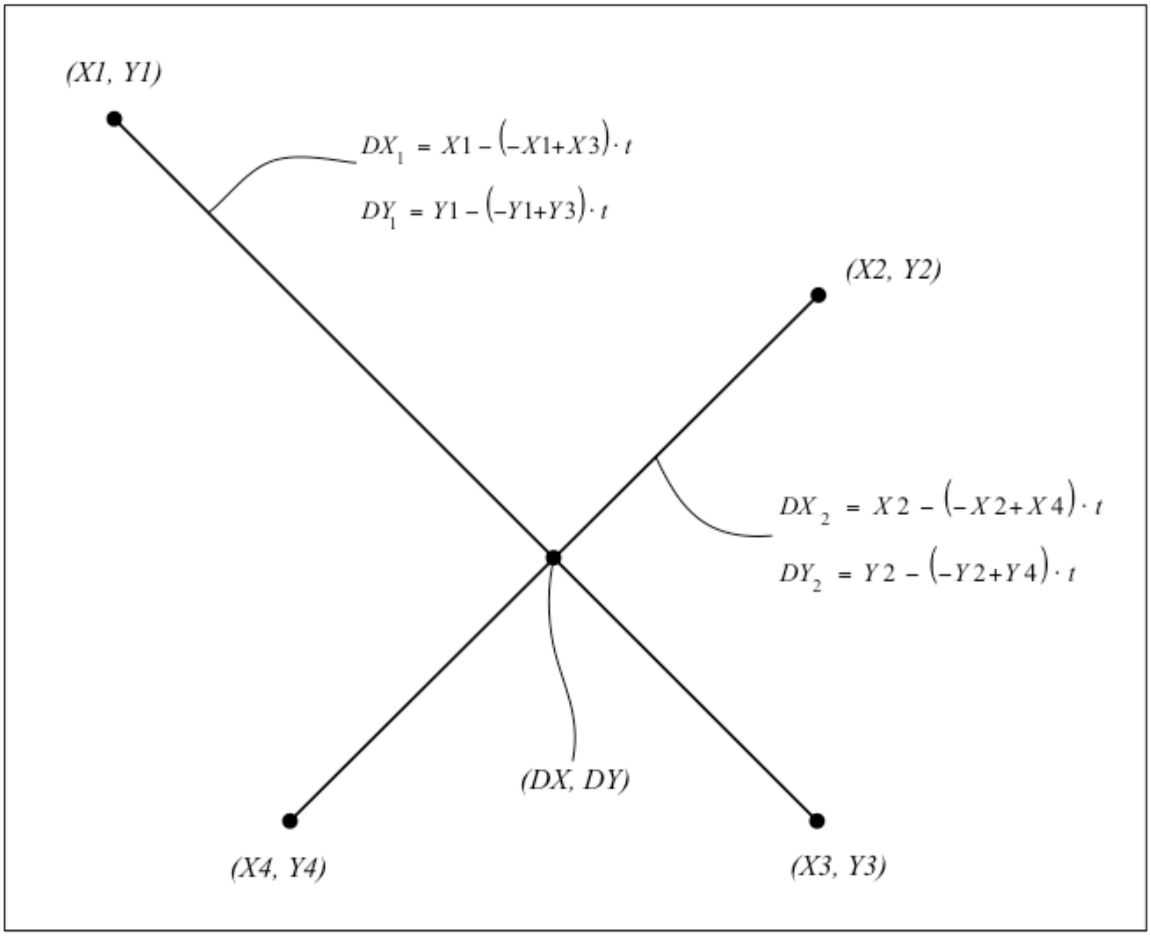

(X1, Y1)
$DX_1 = X1 - (-X1+X3) \cdot t$
$DY_1 = Y1 - (-Y1+Y3) \cdot t$
(X2, Y2)
$DX_2 = X2 - (-X2+X4) \cdot t$
$DY_2 = Y2 - (-Y2+Y4) \cdot t$
(DX, DY)
(X4, Y4)
(X3, Y3)

Figure 3.5 Second Step Alignment Visual - Method 2

Start with

$$D1_X = X1 - (-X1 + X3)t \tag{3.8}$$

$$D1_Y = Y1 - (-Y1 + Y3)t \tag{3.9}$$

and solve for **D1_Y**.

$$t = \frac{D1_X - X1}{X1 - X3} \tag{3.10}$$

$$D1_Y = Y1 - (-Y1 + Y3)\left(\frac{D1_X - X1}{X1 - X3}\right) \tag{3.11}$$

$$D1_Y = Y1 + \left(\frac{Y1 - Y3}{X1 - X3}\right) D1_X - \left(\frac{Y1 - Y3}{X1 - X3}\right) X1 \tag{3.12}$$

$$D1_Y = \left(\frac{Y1 - Y3}{X1 - X3}\right) D1_X + Y1 - \left(\frac{Y1 - Y3}{X1 - X3}\right) X1 \tag{3.13}$$

Equation (3.13) is the equation of the line between points 1 and 3.

Then us

$$D2_X = X2 - (-X2 + X4)t \tag{3.14}$$

$$D2_Y = Y2 - (-Y2 + Y4)t \tag{3.15}$$

and solve for **$D2_Y$**.

$$t = \frac{D2_X - X2}{X2 - X4} \tag{3.16}$$

$$D2_Y = Y2 - (-Y2 + Y4)\left(\frac{D2_X - X2}{X2 - X4}\right) \tag{3.17}$$

$$D2_Y = Y2 + \left(\frac{Y2 - Y4}{X2 - X4}\right) D2_X - \left(\frac{Y2 - Y4}{X2 - X4}\right) X2 \tag{3.18}$$

$$D2_Y = \left(\frac{Y2 - Y4}{X2 - X4}\right) D2_X + Y2 - \left(\frac{Y2 - Y4}{X2 - X4}\right) X2 \tag{3.19}$$

Equation (3.19) is the equation of the line between points 2 and 4.

Then set

$$D1_Y = D2_Y = D_Y \tag{3.20}$$

and

$$D1_X = D2_X = D_X \tag{3.21}$$

and solve to get the X and Y-values of the center point of the laser plate shown by **D_X** and **D_Y** in equations (3.24) and (3.25) respectively.

$$\left(\frac{Y1-Y3}{X1-X3}\right)D_X + Y1 - \left(\frac{Y1-Y3}{X1-X3}\right)X1 = \left(\frac{Y2-Y4}{X2-X4}\right)D_X + Y2 - \left(\frac{Y2-Y4}{X2-X4}\right)X2 \tag{3.22}$$

$$D_X\left(\left(\frac{Y1-Y3}{X1-X3}\right) - \left(\frac{Y2-Y4}{X2-X4}\right)\right) = Y2 - Y1 - \left(\frac{Y2-Y4}{X2-X4}\right)X2 + \left(\frac{Y1-Y3}{X1-X3}\right)X1 \tag{3.23}$$

$$D_X = \frac{Y2-Y1-\left(\frac{Y2-Y4}{X2-X4}\right)X2+\left(\frac{Y1-Y3}{X1-X3}\right)X1}{\left(\frac{Y1-Y3}{X1-X3}\right)-\left(\frac{Y2-Y4}{X2-X4}\right)} \tag{3.24}$$

$$D_Y = \left(\frac{Y1-Y3}{X1-X3}\right)\left(\frac{Y2-Y1-\left(\frac{Y2-Y4}{X2-X4}\right)X2+\left(\frac{Y1-Y3}{X1-X3}\right)X1}{\left(\frac{Y1-Y3}{X1-X3}\right)-\left(\frac{\square 2-Y4}{X2-X4}\right)}\right) + Y1 - \left(\frac{Y1-Y3}{X1-X3}\right)X1 \tag{3.25}$$

Then use the input value **alphaR** calculated by the code block in the main control program to solve for **moveX5** and **moveY5** shown in equations (3.30) and (3.31) respectively.

$$RmoveX5 = D_X - 25 \tag{3.26}$$

$$RmoveY5 = D_Y - 25 \tag{3.27}$$

$$alphaQ = atan\left(\frac{Y3-Y1}{X3-X1}\right) - atan\left(\frac{25}{25}\right) \tag{3.28}$$

$$deg1 = alphaQ\left(\frac{180}{\pi}\right) \tag{3.29}$$

$$moveX5 = Rmov\square X5 * \cos(-alphaR) - RmoveY5 * \sin(-alphaR) \tag{3.30}$$

$$moveY5 = RmoveX5 * \sin(-alphaR) + RmoveY5 * \cos(-alphaR) \tag{3.31}$$

which come from the rotation matrix

$$\begin{Bmatrix} moveX5 \\ moveY5 \end{Bmatrix} = \begin{bmatrix} \cos(-alphaR) & -\sin(-alphaR) \\ \sin(-alphaR) & \cos(-alphaR) \end{bmatrix}\begin{Bmatrix} RmoveX5 \\ RmoveY5 \end{Bmatrix} \tag{3.32}$$

where **moveX5** and **moveY5** are the inputs to the robot in the robot coordinate frame for translation movements and **deg1** is the rotational input command in degrees in the tool coordinate frame.

This calculation method is used in a feedback loop to update the positions of the lasers until the center of the laser plate is within an allowable error tolerance of the center of the grid plate. The allowable error tolerance will be discussed with the test results described in Chapter 7.

The system uses three methods to complete the entire alignment process. The position of each laser is calculated using an averaging technique like that used to calculate the 2D center of gravity of a flat plate. The positions are then used to calculate the translation in the X and Y-directions to move the lead laser to the Home Position on the Grid Array. The rotations and translations to align the other three lasers to the Grid Array are then calculated by measuring the positional distance between the center of the laser plate and the center of the Grid Array.

4. SYSTEM COMPONENTS DETAIL

The full alignment system consists of the laser module, the photodiode module, the DAQ, a computer running the control system, and a robot. Each component, except for the computer, is discussed in this chapter. As long as the computer can run LabVIEW and meets basic standard computer specifications, the specifics of the computer are not critical.

The two primary hardware/electrical components of the alignment system are Module A and Module B. Each consists of a plastic base plate with mounted circuit boards that either mount a laser controller PCB board and laser diodes, as does Module A or that mounts a PCB board containing the photodiodes, as does Module B. The components of each module are described in detail below.

Module A consists of four laser diodes and a controller to regulate the voltage input, and therefore, light intensity output of the laser diodes. The controller is powered by a 7 Volt DC supply, and uses the photodiode built into the laser diode package to regulate the voltage to the laser diode. A schematic of the control circuit is shown below in Figure 4.1.

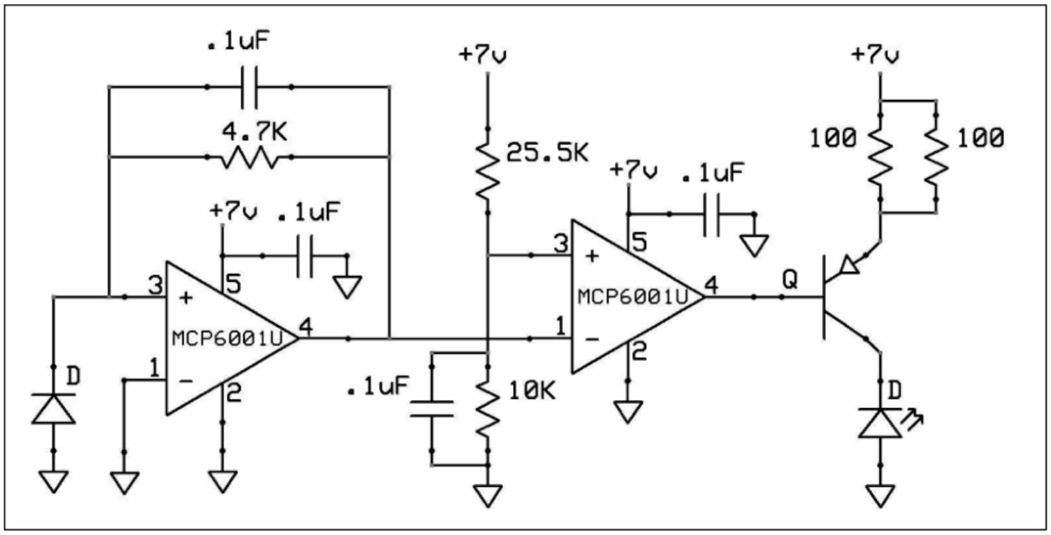


Figure 4.1 Laser Diode Controller Schematic

The controller takes +7 volts in and inserts circuitry between two of the pins of the laser diode. Voltage input to the photodiode is regulated based on the voltage output of the built-in photodiode. The system functions as a feedback loop. The laser diode is supplied with a voltage that causes it to emit light. The light is detected by the built-in photodiode, which produces a voltage output that the controller processes. Based on that voltage output the controller adjusts the voltage input to the laser diode, completing the loop.

The controller board consists of four quadrants, each controlling one laser, drawing on the common power input and connecting to the common ground. Figure 4.2(a) is a graphical layout model and Figure 4.2(b) is a photo of the actual assembled board. The controller works independently from a computer, instead receiving power from a power supply. The controller connects to each laser diode through three wires. The laser diodes are mounted to small boards as shown in Figure 4.3. The boards provide mounts for the lasers, which are then mounted to Module A as show in Figure 4.4. Each laser diode mounting assembly retains a lens (shown in Figure 4.5) to focus the laser spot on the photodiode plate.

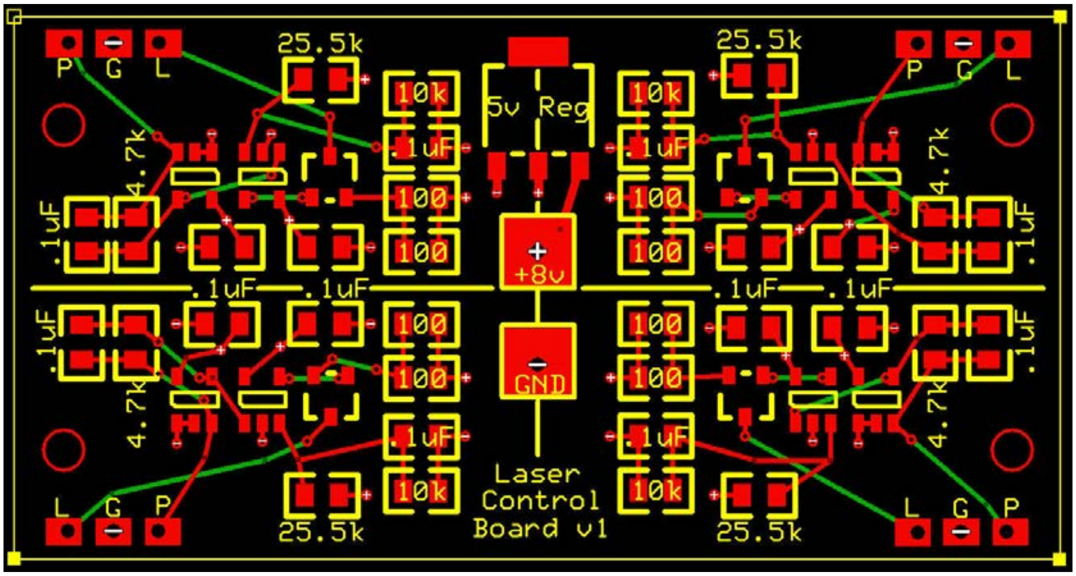


Figure 4.2(a) Module A Control Board Layout

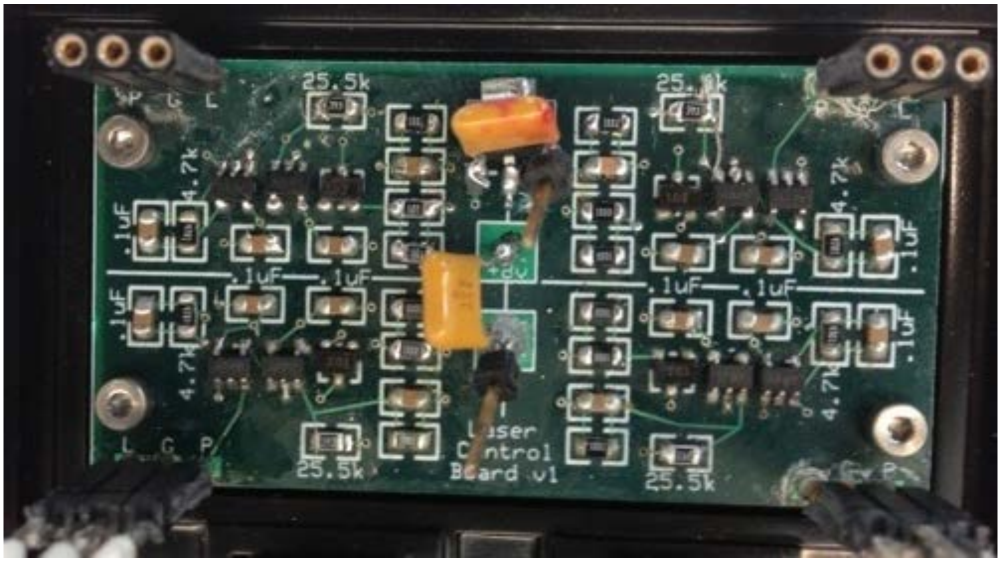


Figure 4.2(b) Module A Control Board Photo

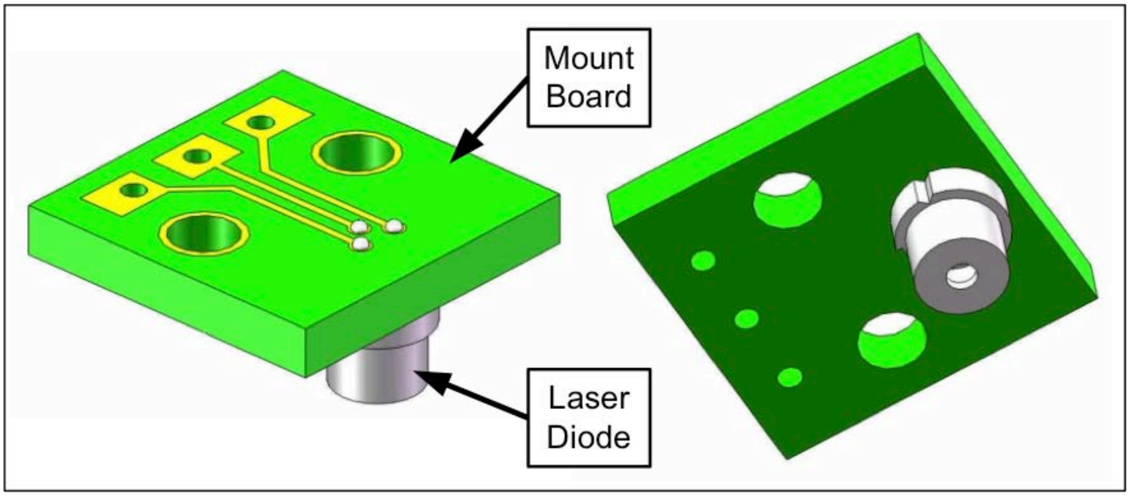


Figure 4.3 Laser Diode Mount and Adapter Board

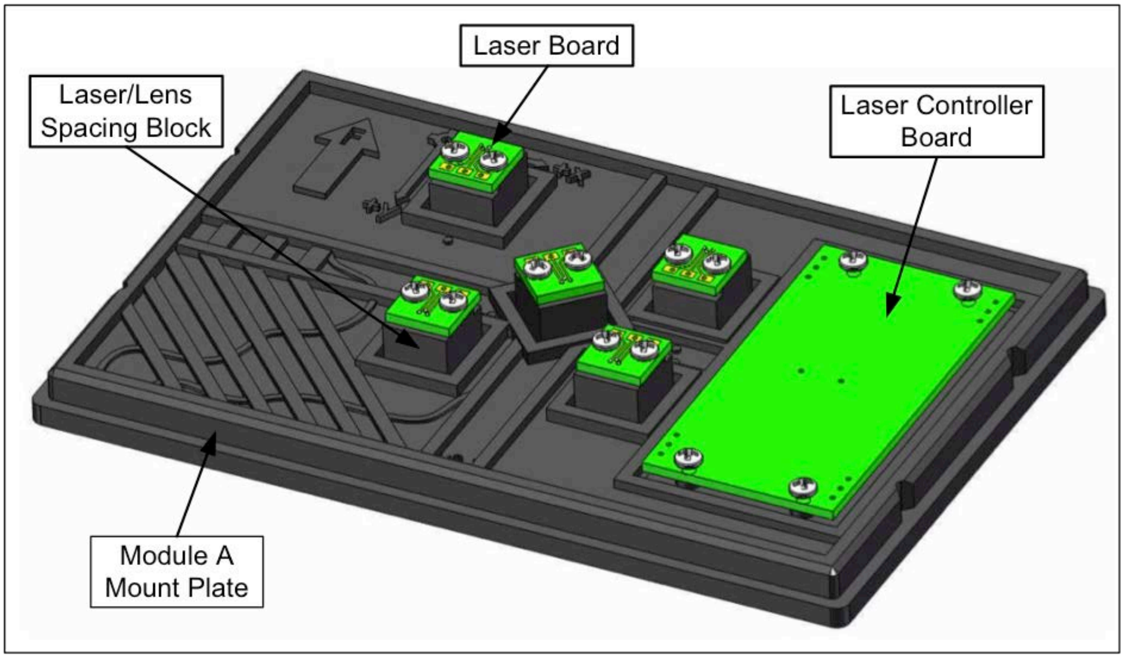


Figure 4.4 Labeled Components of Module A

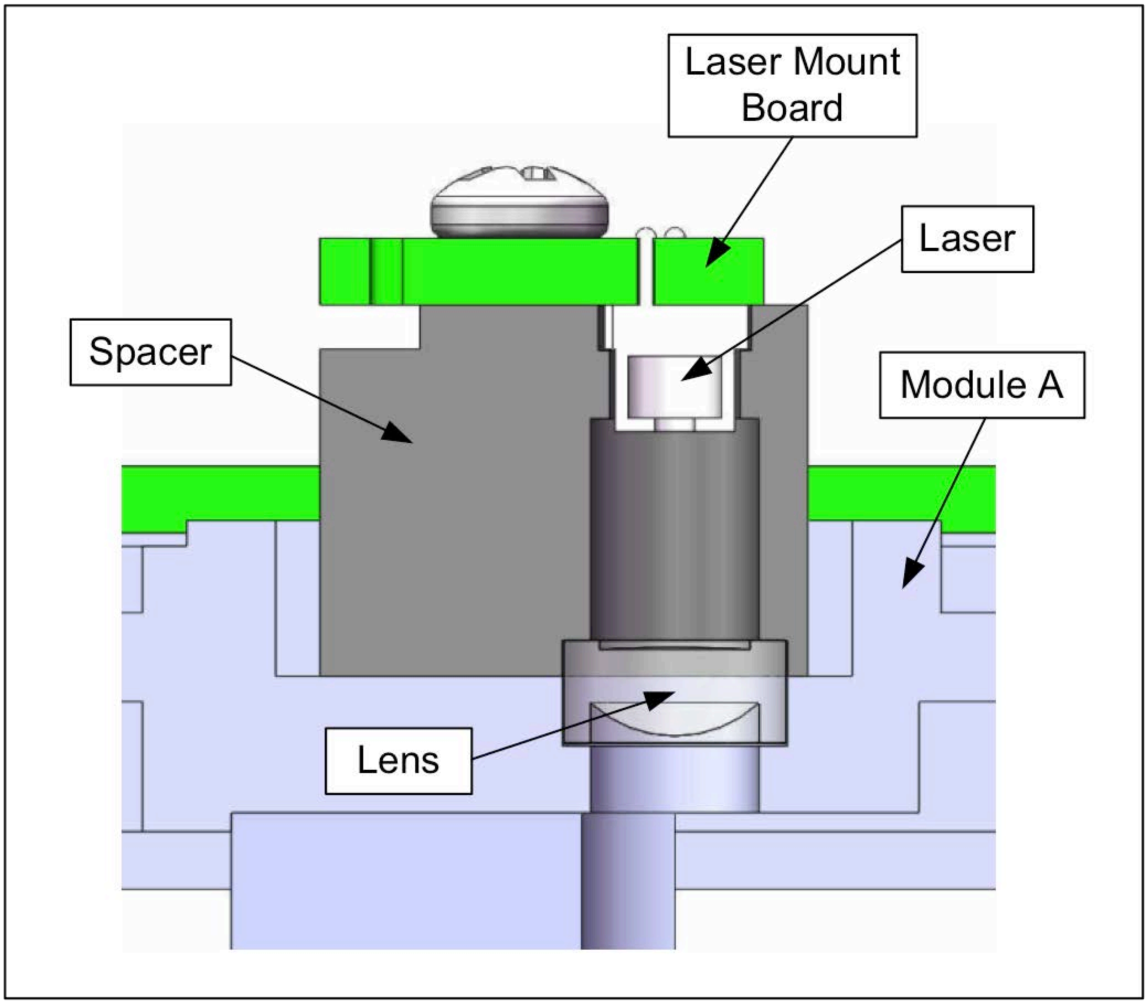


Figure 4.5 Laser Diode and Lens Cross Section

The lasers are small laser diodes of type ADL-65077SU with a diameter of 3.3mm and a height of 3.4mm supplied by Roithner LaserTechnik GmbH. The laser is mounted to a small adapter board that breaks out the wires to route them to the laser controller board and contains mounting holes to mount it to the spacer. The laser body then sits in a mounting feature in the spacer that keeps it a specified distance from a lens that helps *collimate* and focus the laser beam. This spacer is mounted to Module A, which is held by the robot gripper. A cross-section diagram of the laser, lens, and spacer on Module A is shown in Figure 4.5 above. Additionally, a photo of a laser diode next to a penny is shown in Figure 4.6 below to give a size reference for the laser diode and a dimensioned drawing of the laser diode is shown in Figure 4.7.

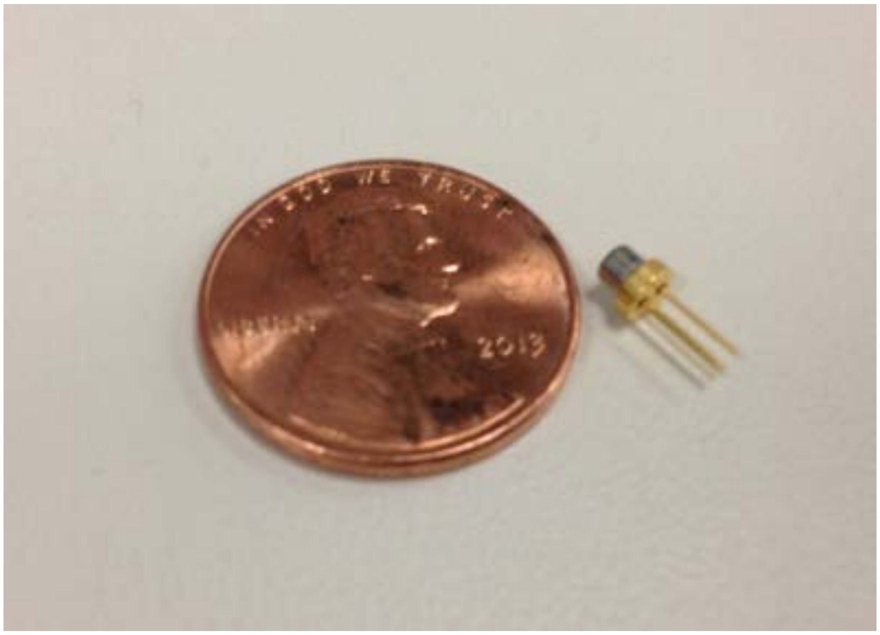


Figure 4.6 Laser Diode Photo With Penny as Reference

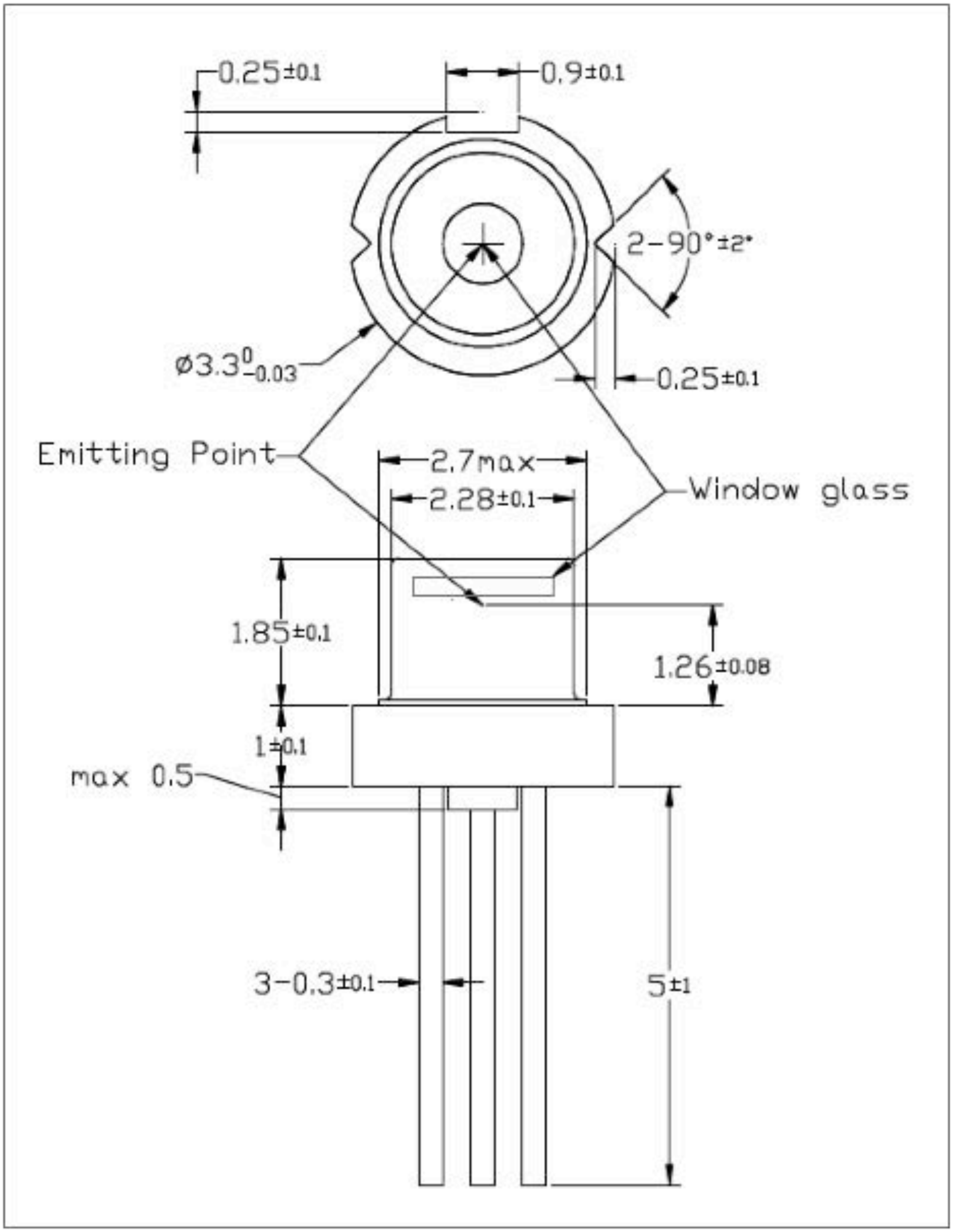


Figure 4.7 Laser Diode Dimensional Drawing [Appendix E-4]

Based on the classifications given by the Lawrence Berkeley National Laboratory affiliated with the U.S. Department of Energy [16] and Rockwell Laser Industries [17], the laser used in the experimental setup of this thesis is classified as a Class 3B laser. Consequentially direct eye exposure is dangerous, but exposure to diffused reflections is not dangerous as long as the output power remains under 500mw. Protective eyewear is recommended as an additional precaution. The laser is in the 645nm to 660nm range and has a maximum continuous wave (CW) light output of 10mW, therefore remaining in the Class 3B specification.

The laser diodes are arranged in a square pattern corresponding to each quadrant of the Grid Array. The lead laser is pulled out along a diagonal from the other three lasers. This allows the system to better detect the orientation of the plate. If the square was increased in size so that the aligned position put each laser on a corner of the Grid Array, then even a small rotation would place two of the lasers on the edge or off the grid entirely. Furthermore, if all four lasers were in the same smaller square as the three lasers, then it would be difficult to perform the first step of the alignment without interference from the other three lasers. They would be close enough to the lead laser to shine light onto photodiodes near the edge of the grid.

Figure 4.8 shows the square of lasers with the lead laser pulled out along the diagonal. Figure 4.9 shows the square of lasers with all four lasers on the corners of the Grid Array, and Figure 4.10 shows the larger square of lasers as shown in Figure 4.9 but rotated clockwise.

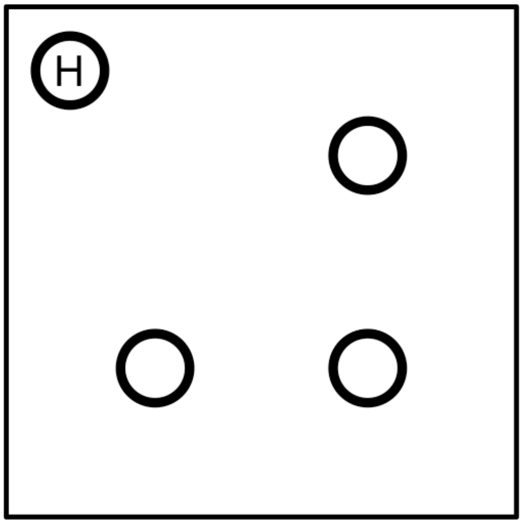


Figure 4.8 Laser Square Arrangement with Lead Laser Pulled out Along Diagonal

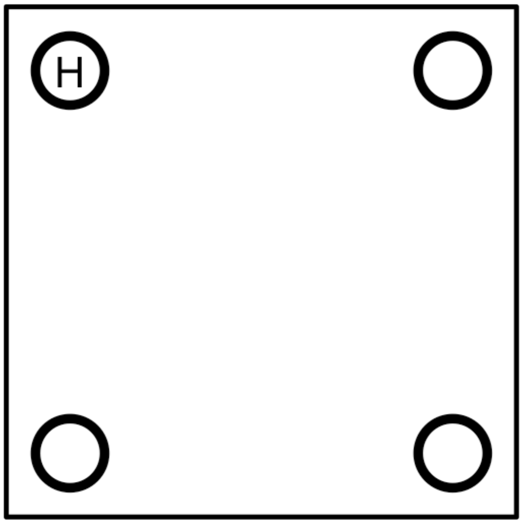


Figure 4.9 Laser Square Arrangement with All Lasers at Grid Corners

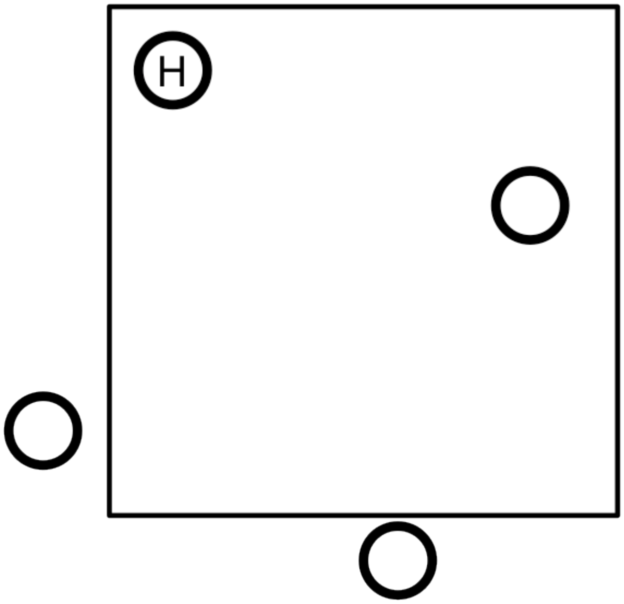


Figure 4.10 Laser Larger Square Arrangement Rotated Clockwise

In the experimental setup, the lasers tend to focus most of the light, while scattering a small amount across a large area. To therefore clean up the laser output and reduce the effect of the scattered light across surrounding photodiodes, cylinders are glued to the underside of Module A as shown in Figure 4.11 to help further collimate the laser spot.

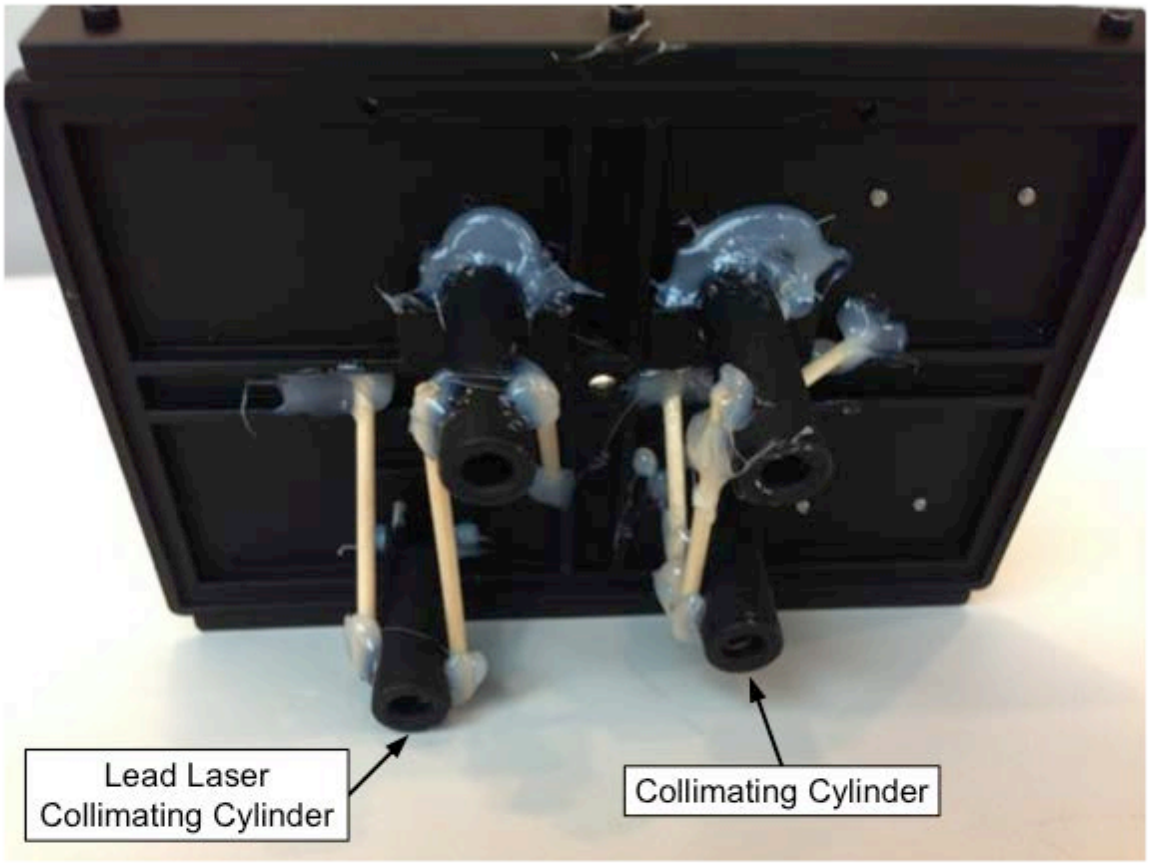


Figure 4.11 Laser Collimating Cylinders on Underside of Module A

The sensors on Module B consist solely of photodiodes each built into an SOT-8 chip that includes an amplification circuit. The amplification circuit is necessary to allow the data acquisition (DAQ) component of the system to sufficiently detect the voltage output of the photodiode. The natural output of a photodiode is in the millivolt range with a small current. In order to transmit a signal that is more immune to transmission noise, the photodiode output must be amplified. This is done in the OPT101 SOT-8 chip by Texas Instruments, Inc.

The original design consisted of the EPD-660-1-0.9 SMD 1206 chip from Roithner LaserTechnik, GmbH with an amplification circuit consisting of a 1.8V, 1MHZ, SOT23-5 op-amp, a 22pF capacitor, and a 50kOhm resistor as shown in Figure 4.12 below. Ultimately the OPT101 was chosen to carry out initial grid-array testing. The

EPD-660-1-0.9 is strictly a photodiode and would require an amplification circuit that would use more space and components than the OPT101 chip. Despite the compact nature of the OPT101 chip, it is not possible to achieve as dense of a Grid Array with this chip. One compensating factor though is that the sensor area of the OPT101 chip is nearly 8.5 times that of the EPD-660-1-0.9, providing an overall larger combined sensing area. See Figure 4.13 for a representation of Module B with the grid-array of the larger photodiodes. Additionally, a photo of a photodiode next to a penny is shown in Figure 4.14.

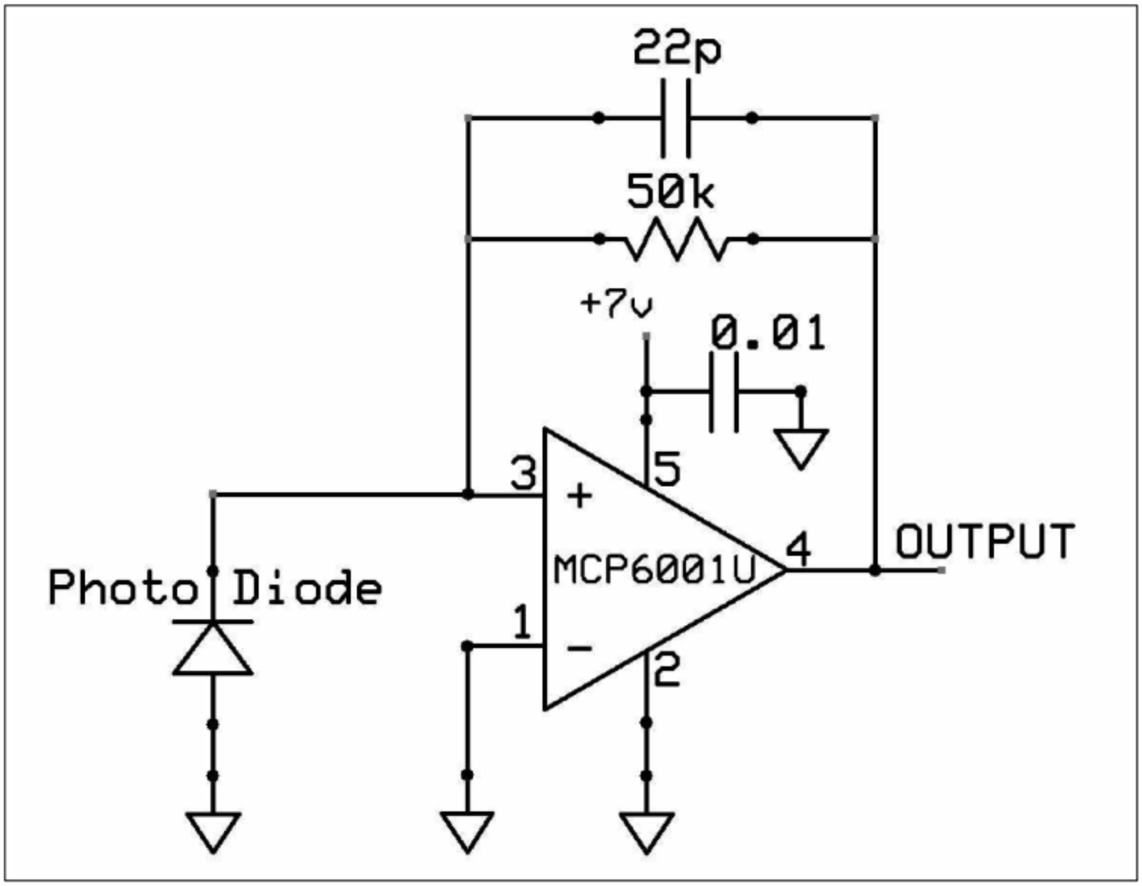


Figure 4.12 Photodiode Amplification Circuit

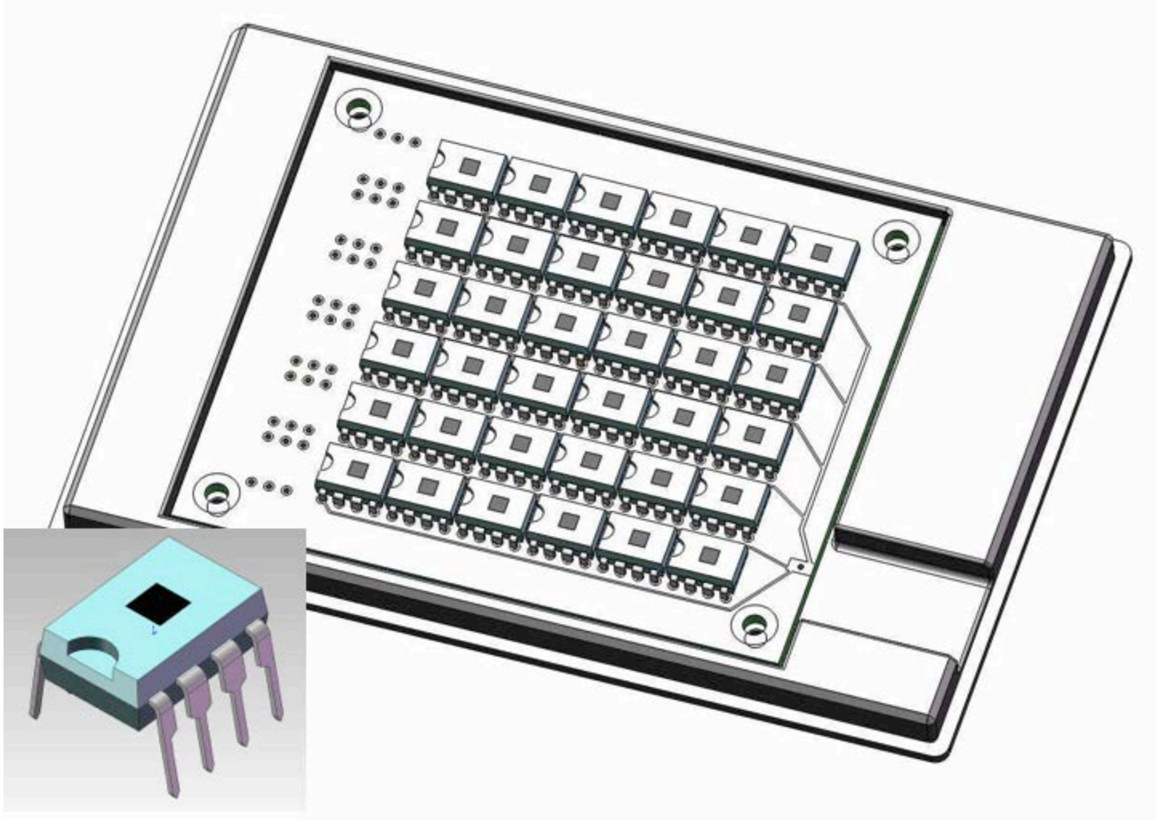


Figure 4.13 Module B with Grid-Array of Photodiodes and OPT101 Photodiode Chip Inset

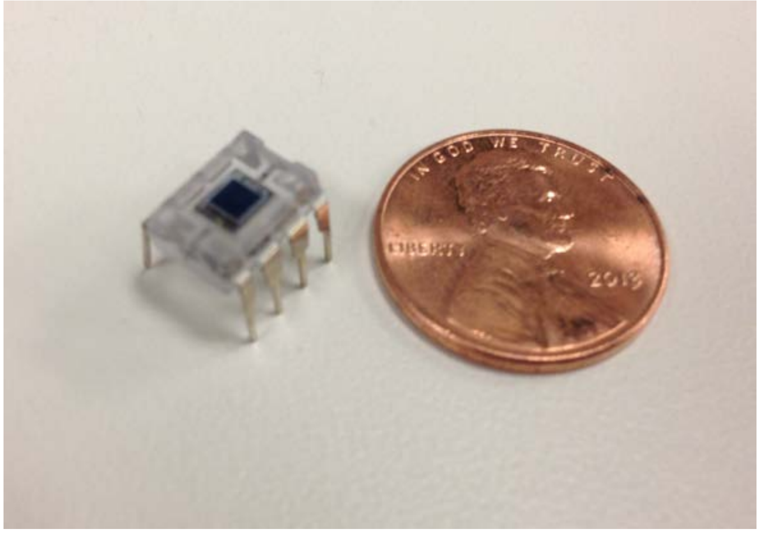


Figure 4.14 Photo of Photodiode With Penny as Reference

The OPT101 photodiode chip is sensitive to light in the bandwidth of 260nm to 1100nm, with peak sensitivity near 840nm. To limit the ambient light entering the photodiode, it is possible to either apply a light filter over the photodiodes, which would permit only light near the output frequency, or to operate the system in a dark setting. Another option is to select a photodiode with a narrow detection band and select a corresponding laser diode. In the testing done for this thesis, a very low lighting was used in addition to a software filter to remove the effect of any remaining ambient light.

Figure 4.15 below shows the size comparison between the photodiode and the laser diode by placing both side by side on the surface of a penny.

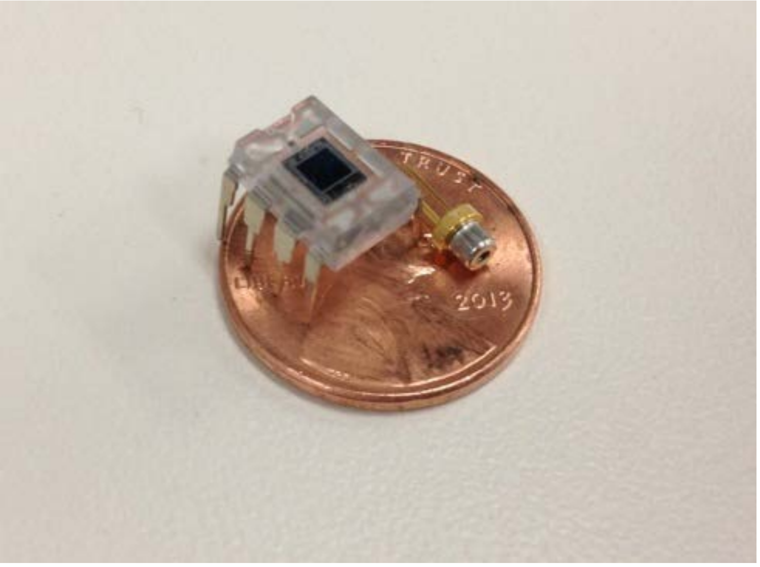


Figure 4.15 Photo of Photodiode and Laser Diode With Penny as Reference

Module A and Module B mounting plates are produced with an SLA rapid prototyping method and spray-painted black. This allows for a quick production of a complex mounting piece as well as rapid recreation to accommodate changes in design. Each module has a custom-designed circuit board mounted to it.

The specific SLA rapid prototyping machine used to produce the Module A and Module B mounting plates is shown below in Figure 4.16.

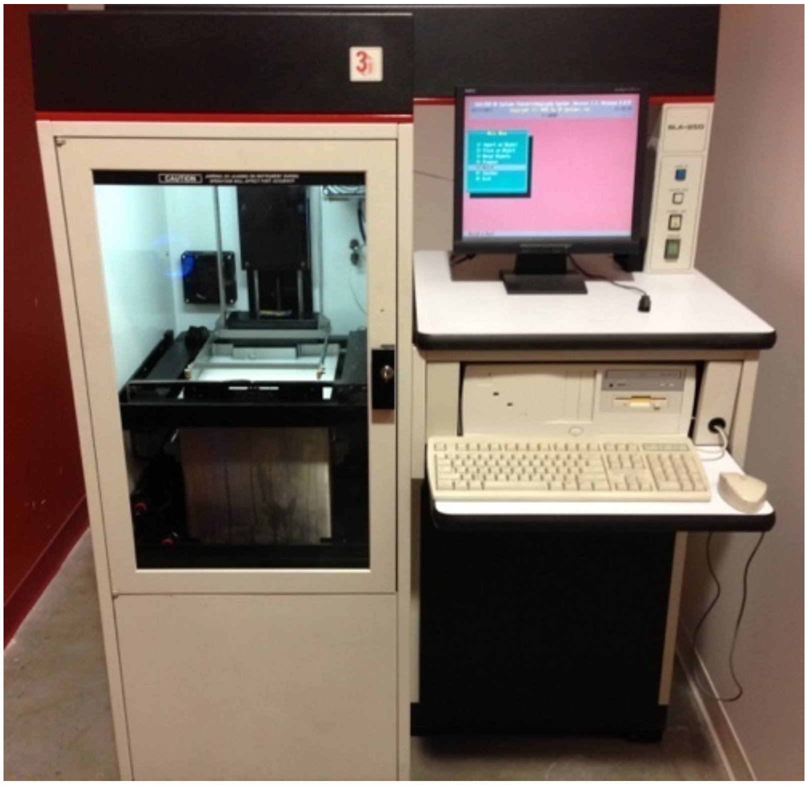


Figure 4.65 SLA Rapid Prototyping Machine Used To Create Module A and Module B Plastic Components

The DAQ is composed of analog-to-digital boards, which are Arduino MEGE 2560 microcontroller boards (See Figure 4.17). Each board accepts sixteen analog inputs (twelve of which are used on each board for this application) and then transmits a resulting digital signal to the computer through a USB cable. To provide enough inputs for this test, three boards were used. They serve as an inexpensive data input device, with each board costing approximately $50.

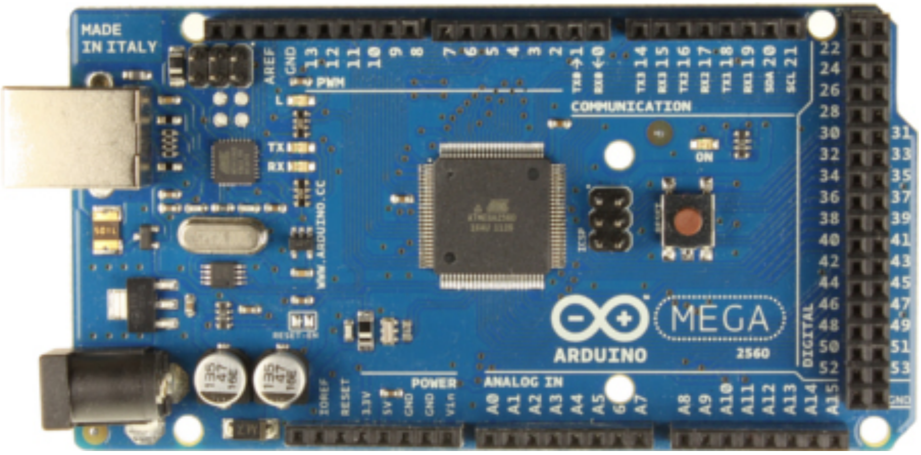


Figure 4.17 Top side of Arduino MEGA 2560 Micro Controller Board [Appendix E-5]

After the commands are created, they are sent to the robot for execution. The robot used in this scenario is a SCARA KiNEDx KX-300x660 robot developed by Peak Robotics, Inc. It sits on a linear rail and consists of a rotational "shoulder" joint, a vertical Z-axis rail, an extending arm consisting of three links and three joints that work in unison (see Figure 4.18 below), and a wrist joint that includes a gripper.

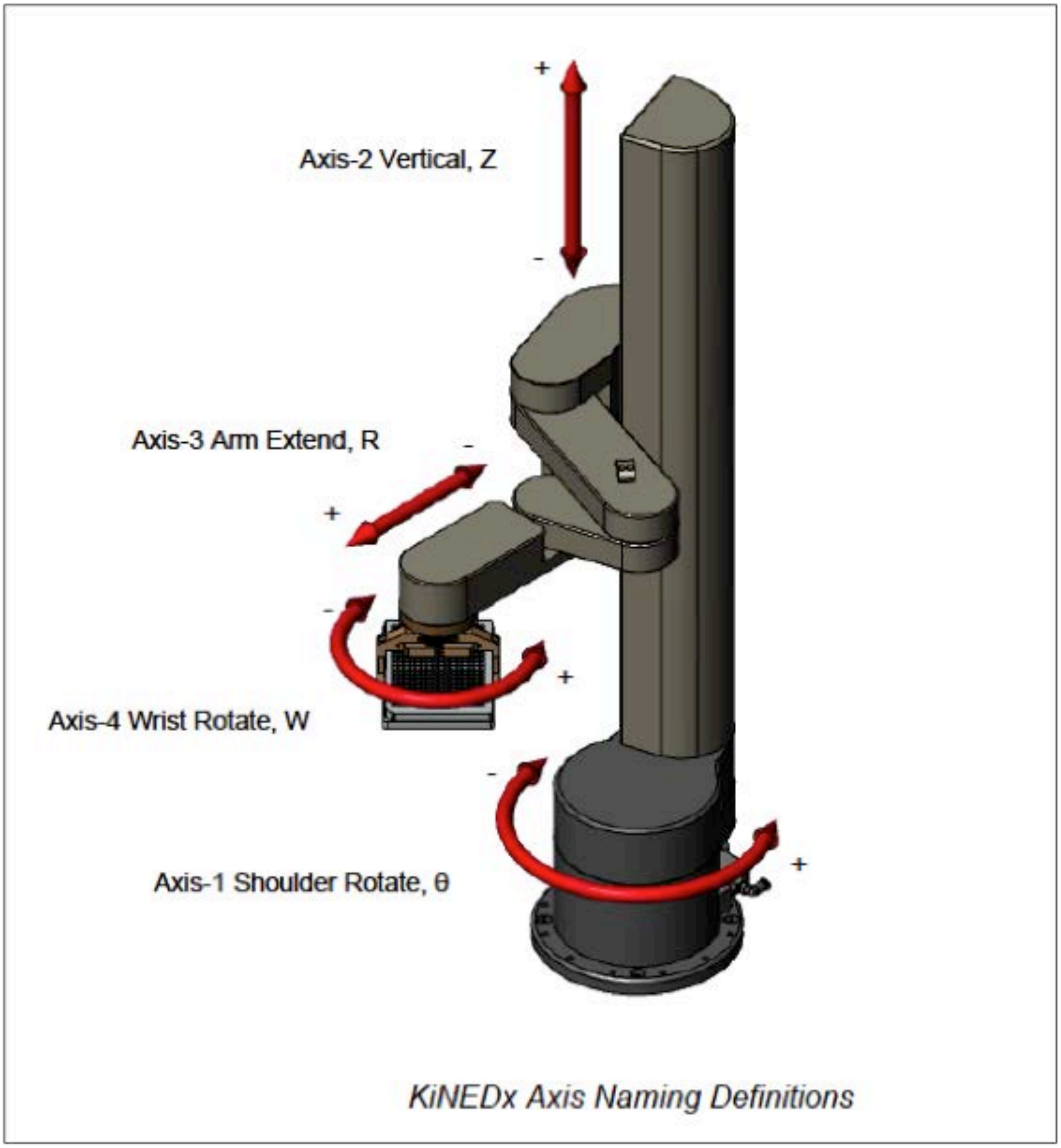


Figure 4.18 Robot Joint and Axis Diagram [Appendix E-6]

Figure 4.19 shows the World Reference Frame of the robot. As can be seen, the coordinates are based on the robot base. Alternatively, in Figure 4.20 it can be seen that the coordinates are relative to the tool. This is the Tool Reference Frame.

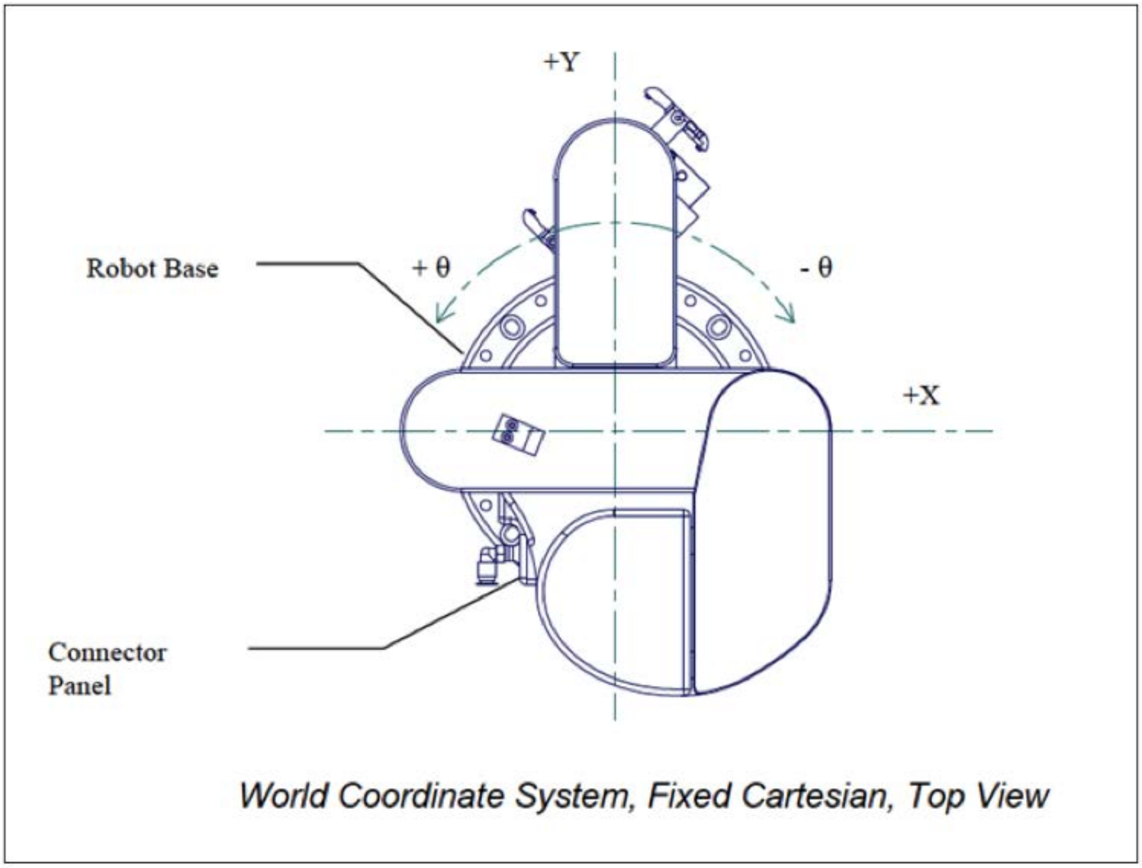


Figure 4.19 Robot Reference Frame [Appendix E-7]

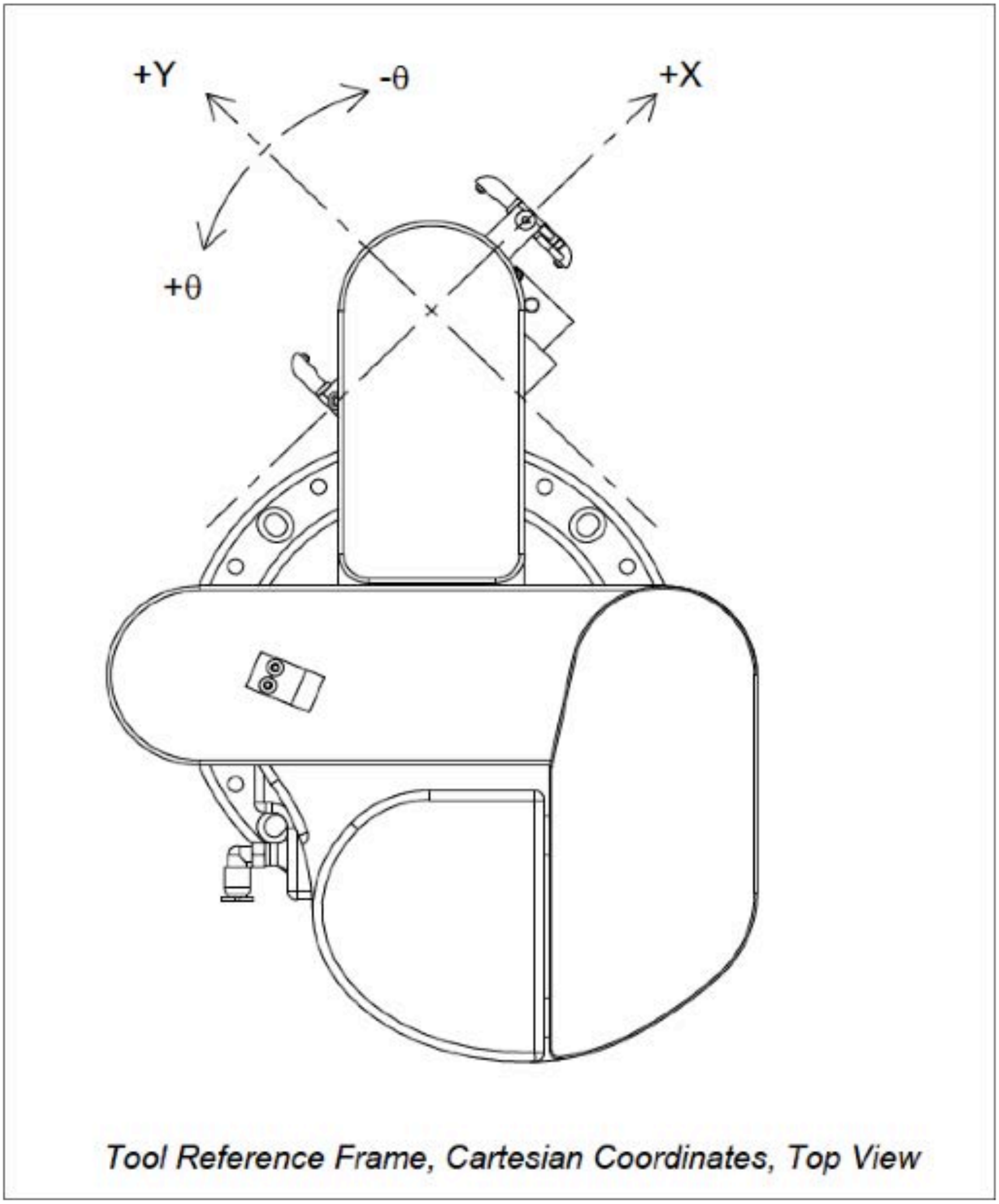
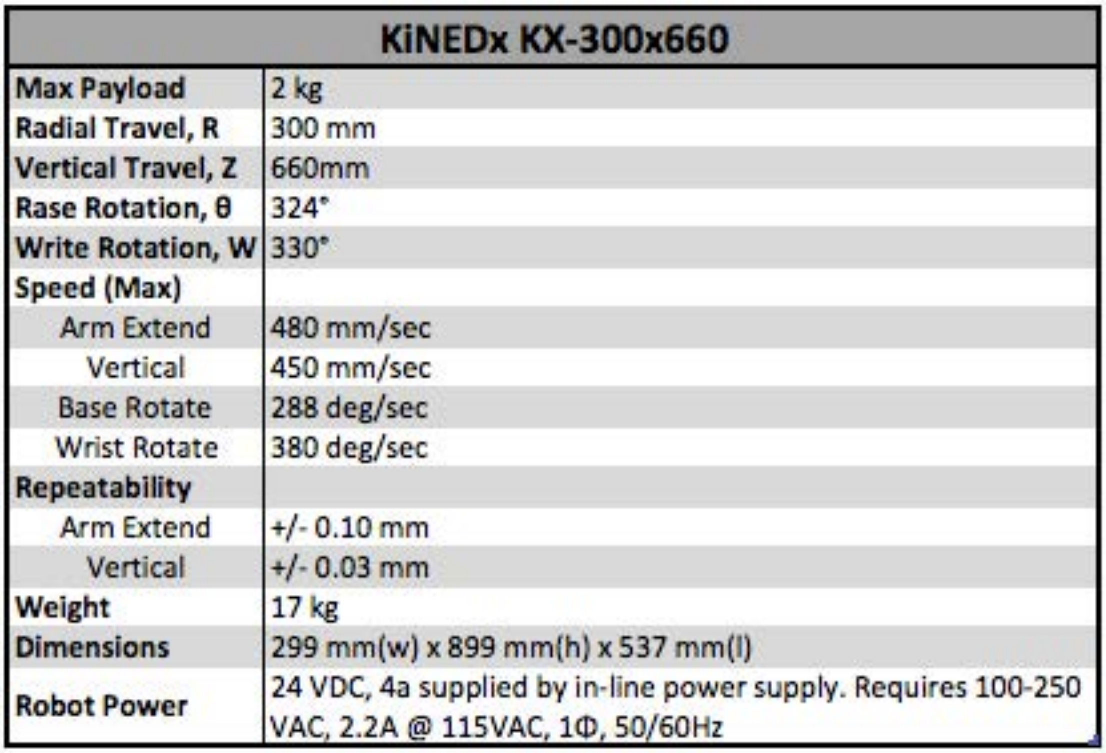


Figure 4.20 Tool Reference Frame [Appendix E-8]

The below Table 4.1 is adapted from the User Manual of the KiNEDx KX-330x660 [17]. It lists multiple parameters of the robot such as maximum travel along the rotational and Z-axes and joint velocity.

Table 4.1 Robot Specifications



KiNEDx KX-300x660	
Max Payload	2 kg
Radial Travel, R	300 mm
Vertical Travel, Z	660mm
Rase Rotation, θ	324°
Write Rotation, W	330°
Speed (Max)	
Arm Extend	480 mm/sec
Vertical	450 mm/sec
Base Rotate	288 deg/sec
Wrist Rotate	380 deg/sec
Repeatability	
Arm Extend	+/- 0.10 mm
Vertical	+/- 0.03 mm
Weight	17 kg
Dimensions	299 mm(w) x 899 mm(h) x 537 mm(l)
Robot Power	24 VDC, 4a supplied by in-line power supply. Requires 100-250 VAC, 2.2A @ 115VAC, 1Φ, 50/60Hz

5. CONTROL SYSTEM DESIGN

The control system consists of the three primary components shown in Figure 5.1: the data acquisition unit (DAQ), the voltage to position converter (VPC), and the position to robotic movement converter (PRC). The DAQ serves as an input point for the voltage signals coming from the photodiodes. The signals are received as analog inputs to allow the intensity of the light hitting each photodiode to be recorded, instead of digital signals that would show which photodiodes detected light, but not how much. In this specific embodiment, the DAQ is a set of three Arduino MEGA 2560 boards each accepting analog inputs as discussed in the previous chapter. The DAQ converts the input voltages to a digital signal that can be transferred via a USB cable to the computer and read in by LabVIEW. The software parses the signal to recognize the individual photodiode readings and can manipulate or analyze them individually.

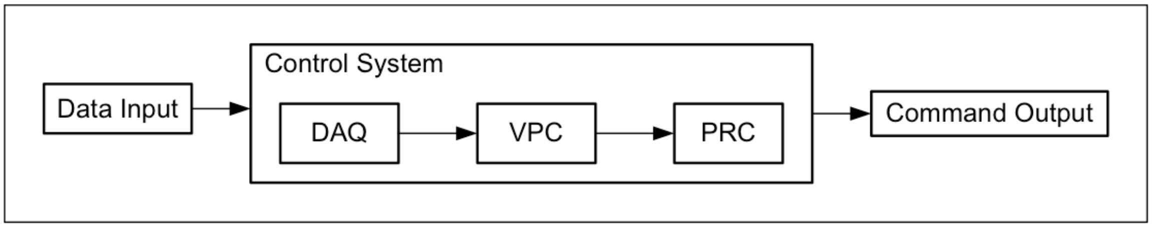


Figure 5.1 Control System High Level Overview

Once the software takes in the signals coming from each input of the DAQ, it splits them two ways. It first sends them to a graphic display that provides a visual indication of the location of the laser beam point on Module B. Next, it sends the signal through a conditioning algorithm that takes in the signal coming from each channel of the DAQ and sets anything below a specified voltage threshold to zero. The purpose of this threshold is to eliminate the signals coming from photodiodes detecting only ambient light. From there the voltage coming from each photodiode is multiplied independently by the distance in millimeters of the photodiode from the Home Position. This creates two sets of data, one relating to the X-direction and one to the Y-direction. The elements of the X-direction set are then summed and divided by the summed voltage outputs of all the photodiodes (also referred to as light intensity of the photodiodes). Furthermore, the same process is performed for the Y-direction set. From these operations the resulting distance is equal to the distance from the laser point to the Home Position. The coordinates of the laser based on the Home Position are then used to develop motion commands to move the robot to the designated position.

The voltage to position converter (VPC) consists of two parts. One part is used for detecting a single laser and produces measurements using the entire grid. The second part detects up to four lasers. It breaks the grid into four equal quadrants, each of which can detect a laser and produce a measurement based on the origin reference point. In both parts of the VPC, a formula similar to that used to calculate a 2D center of gravity is utilized to calculate the position of either the single laser or all four lasers. The voltage from each photodiode is multiplied by the X and Y-coordinates of that photodiode. The sum of the resulting multiplication of each photodiode is also summed and then divided by the summed voltage outputs of all the photodiodes. This is done independently for the X-coordinates and the Y-coordinates to find the laser center. The equations for the X and Y-positions based on the origin point are shown below as equations (5.1) through (5.4). $\boldsymbol{I_n}$ is the voltage reading of each photodiode and $\boldsymbol{I}$ is the summed voltage reading of all photodiodes of the Grid Array.

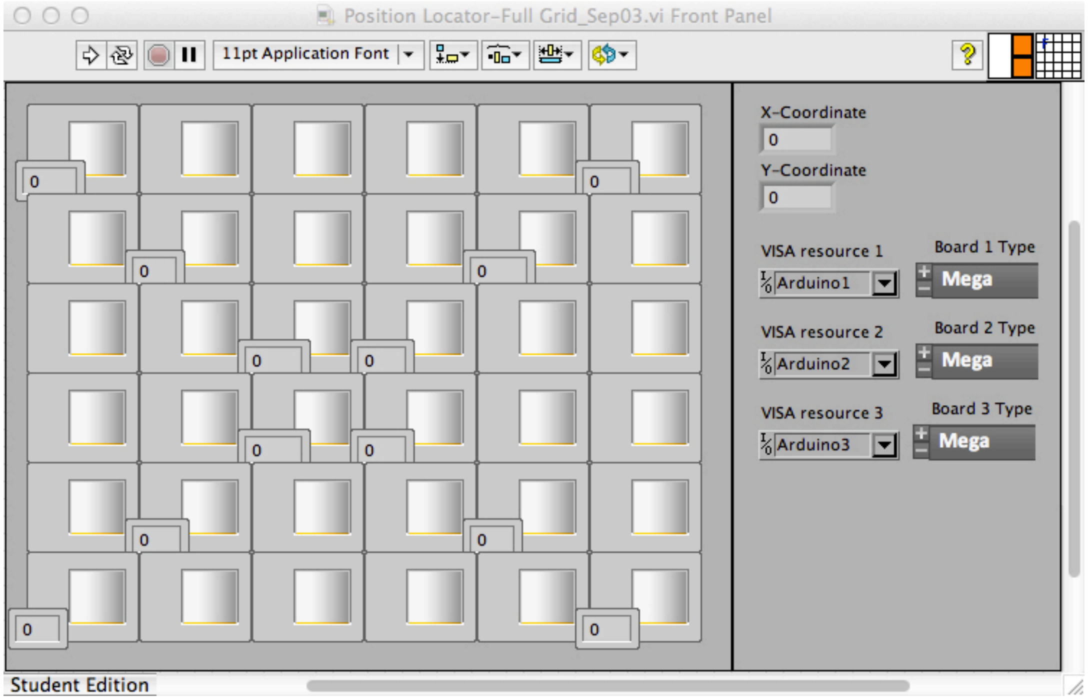
$$XI = x_1 I_1 + x_2 I_2 + \cdots + x_n I_n \tag{5.1}$$

$$YI = y_1I_1 + y_2I_2 + \cdots + y_nI_n \tag{5.2}$$

$$X = \frac{x_1I_1 + x_2I_2 + \cdots + x_nI_n}{I} \tag{5.3}$$

$$Y = \frac{y_1I_1 + y_2I_2 + \cdots + y_nI_n}{I} \tag{5.4}$$

The VPC is embodied by two National Instrument VI's (programs). Each VI is called by the main program as needed. The programs are developed in LabVIEW, which is primarily a graphical programming interface that allows many scientific testing and equipment integration operations to be performed easily. The program files that it produces are called VI's (Virtual Instruments). Each one has a user interface called the **Front Panel** and a graphical code building space called the **Block Diagram**. One VI is called when the entire grid is used to find one point. The other VI is called to detect using a quadrant-discretized grid. Two screen shots of each VI are shown below. The full-grid VI is shown in Figure 5.2 and Figure 5.3 and the quadrant-grid VI is shown in Figure 5.4 and Figure 5.5. The first figure of each set below shows the Front Panel of each VI, while the second figure shows the Block Diagram.



X-Coordinate
0

Y-Coordinate
0

VISA resource 1: Arduino1
Board 1 Type: Mega

VISA resource 2: Arduino2
Board 2 Type: Mega

VISA resource 3: Arduino3
Board 3 Type: Mega

Figure 5.2 Full-Grid Single Laser Detection Front Panel

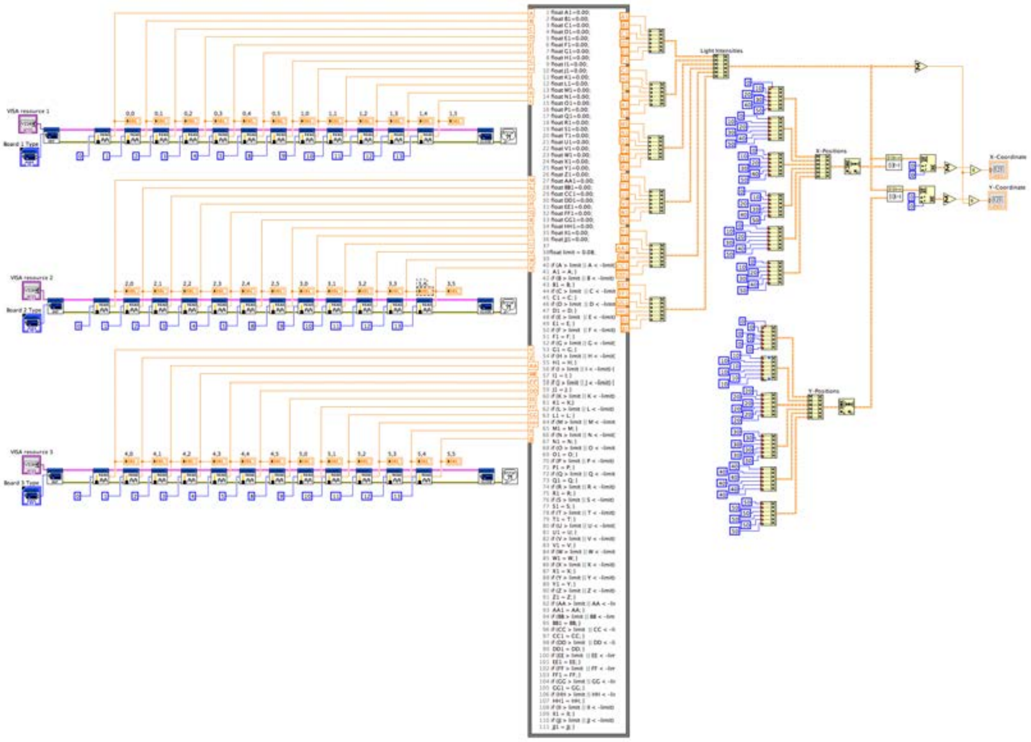
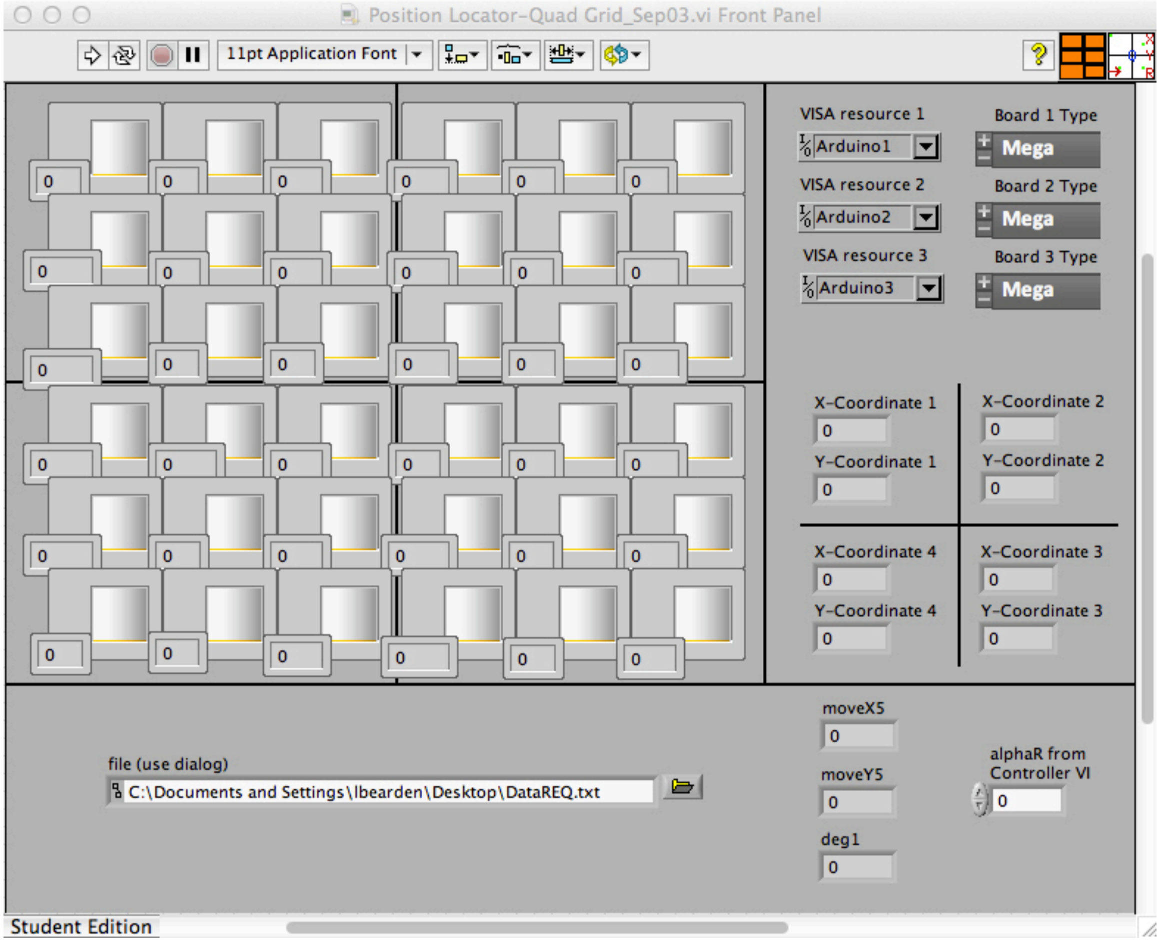


Figure 5.3 Full-Grid Single Laser Detection Block Diagram



0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

VISA resource	Board Type
VISA resource 1: Arduino1	Board 1 Type: Mega
VISA resource 2: Arduino2	Board 2 Type: Mega
VISA resource 3: Arduino3	Board 3 Type: Mega

X-Coordinate 1: 0
Y-Coordinate 1: 0
X-Coordinate 2: 0
Y-Coordinate 2: 0
X-Coordinate 4: 0
Y-Coordinate 4: 0
X-Coordinate 3: 0
Y-Coordinate 3: 0

file (use dialog): C:\Documents and Settings\lbearden\Desktop\DataREQ.txt

moveX5: 0
moveY5: 0
deg1: 0
alphaR from Controller VI: 0

Figure 5.4 Quadrant-Grid Four Laser Detection Front Panel

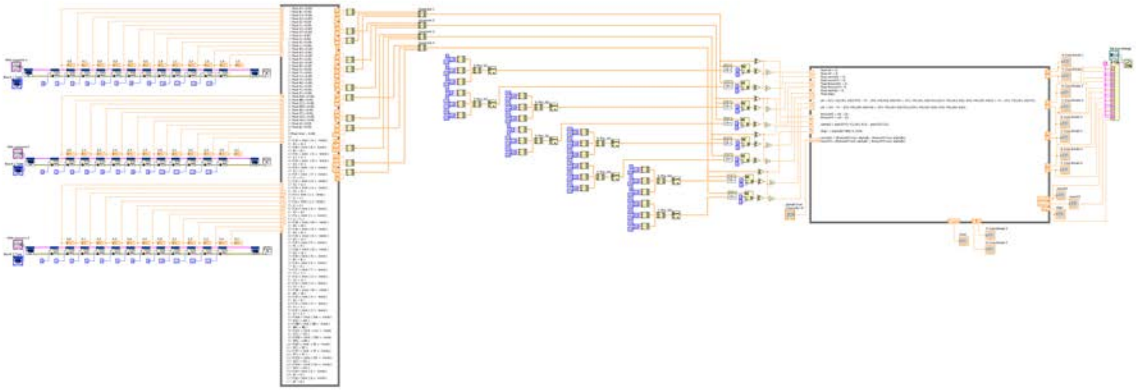


Figure 5.5 Quadrant-Grid Four Laser Detection Block Diagram

The program that receives the positional data from the two position detection VI's is another LabVIEW VI. This one, shown in Figure 5.6, receives X, Y, and rotation values from the other two programs. Using these values, it calculates the necessary movements of the gripper to align Module A and Module B. A detailed description of the operation of the main program follows. Figure 5.7 shows the user interface of the main control program.

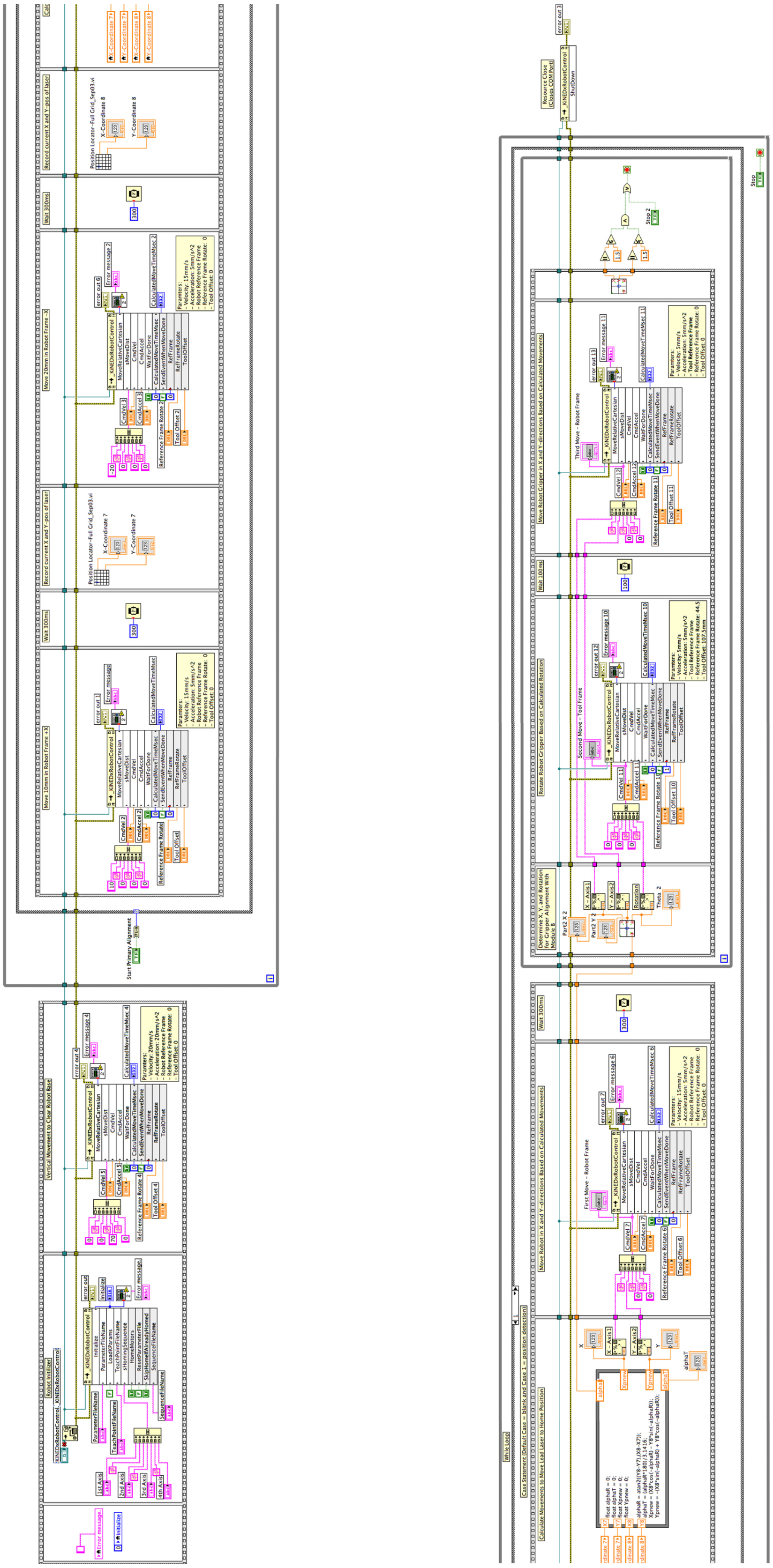
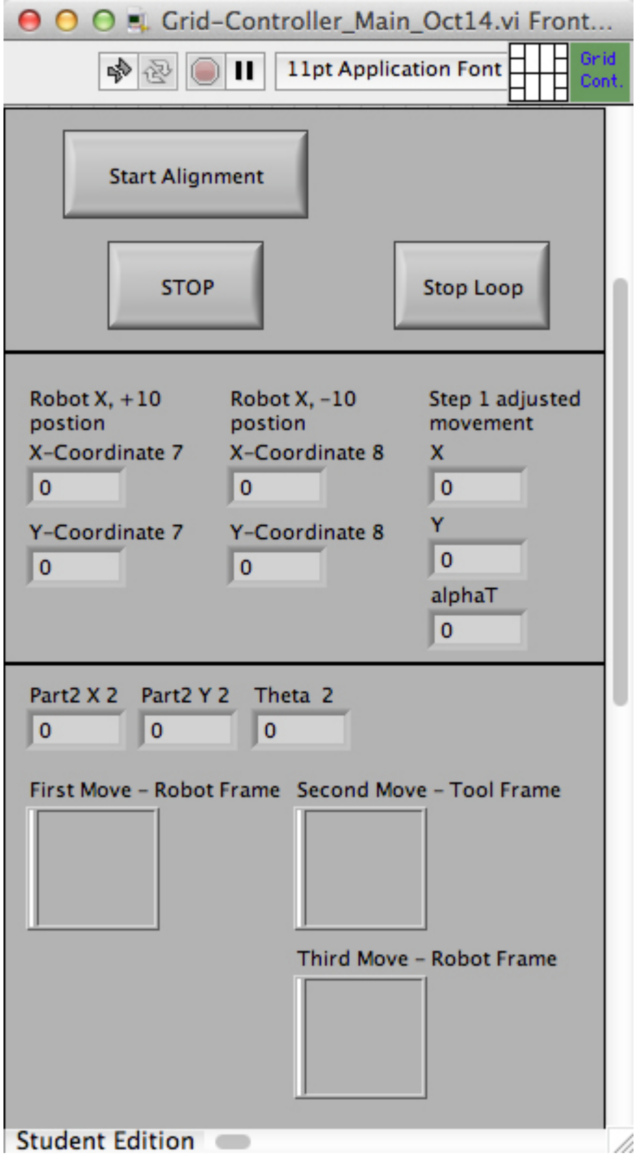


Figure 5.6 Control System Main Program



Robot X, +10 postion

X-Coordinate 7: 0

Y-Coordinate 7: 0

Robot X, -10 postion

X-Coordinate 8: 0

Y-Coordinate 8: 0

Step 1 adjusted movement

X: 0

Y: 0

alphaT: 0

Part2 X 2: 0

Part2 Y 2: 0

Theta 2: 0

First Move - Robot Frame

Second Move - Tool Frame

Third Move - Robot Frame

Figure 5.7 Control System Main program User Interface

Before describing all details of the LabVIEW program, it is essential to understand the high-level functionality of the program. Figure 5.8 represents the flow of the program. The upper flow represents the high-level process of the computer controller. The breakout section describes what happens primarily in the control program shown in Figure 5.6 above.

The photodiodes output analog signals that are converted to digital signals by the DAQ and passed into the computer. Based on the position of the laser at various times, a multi-step alignment process is performed. The first step is to move 10mm along the positive X-axis of the robot and detect the current position, then move the gripper (holding the laser) back 20mm along the negative X-axis of the robot coordinate frame. This second point is then recorded. From these two points, the X-vector of the robot space is calculated to determine the relationship between the grid coordinate frame and the robot coordinate frame. Once the lead laser has been moved into place using commands formed based on the previous relationship, the rotational misalignment between the two plates is calculated using the lead laser along with the other three lasers. Based on the output of the calculations, the robot is moved into its final location.

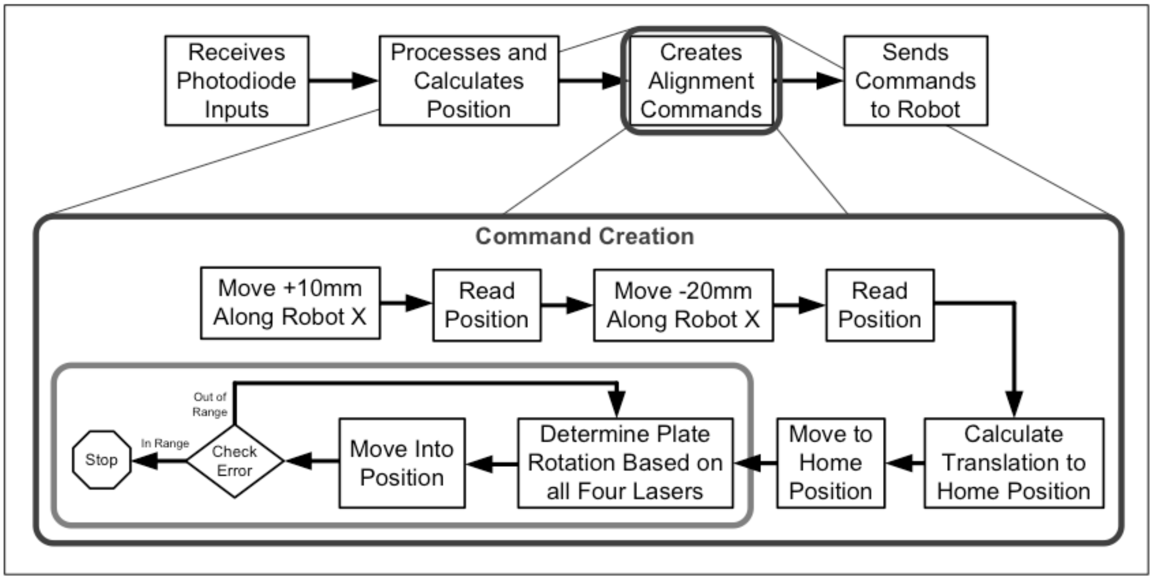


Figure 5.8 Control System High-Level Block Diagram

The program works similarly to programs written in a text-based code, following a sequential progression. Figure 5.9 shows the component of the code that receives input from the DAQ. The module labeled INT receives the data and reads each channel from the DAQ independently. The data is then sent to a viewing screen as shown in Figures 5.2

and 5.4. It is also sent into a text-based code block to be filtered and then placed into a vector as shown in Figure 5.10.

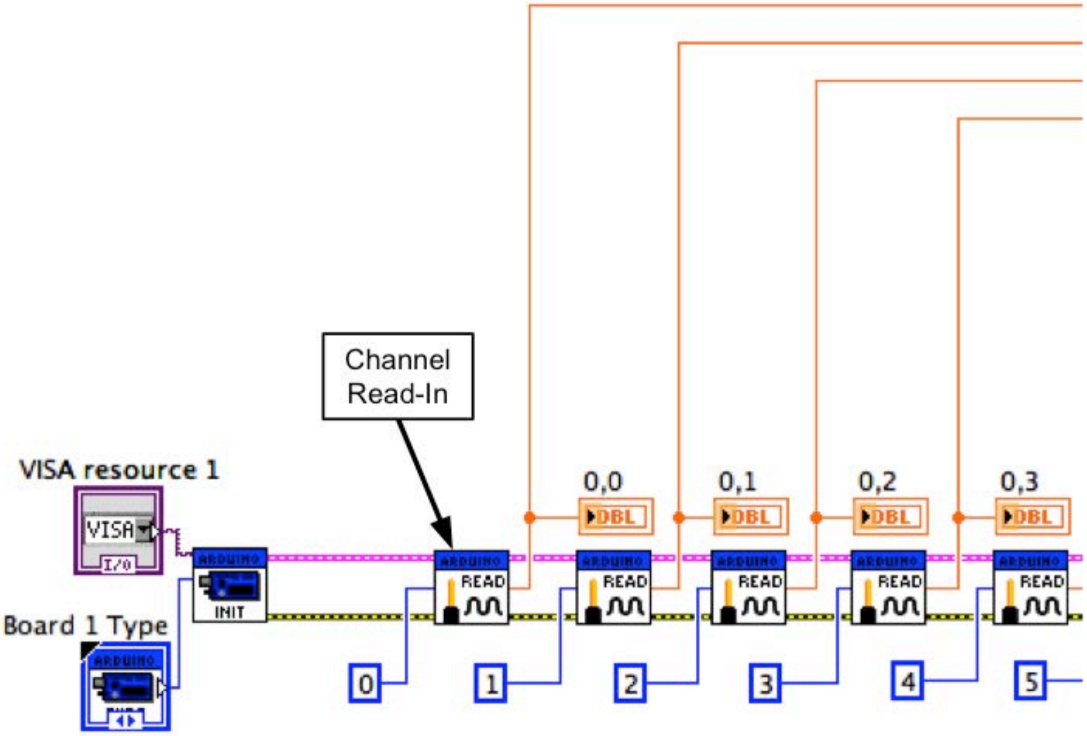
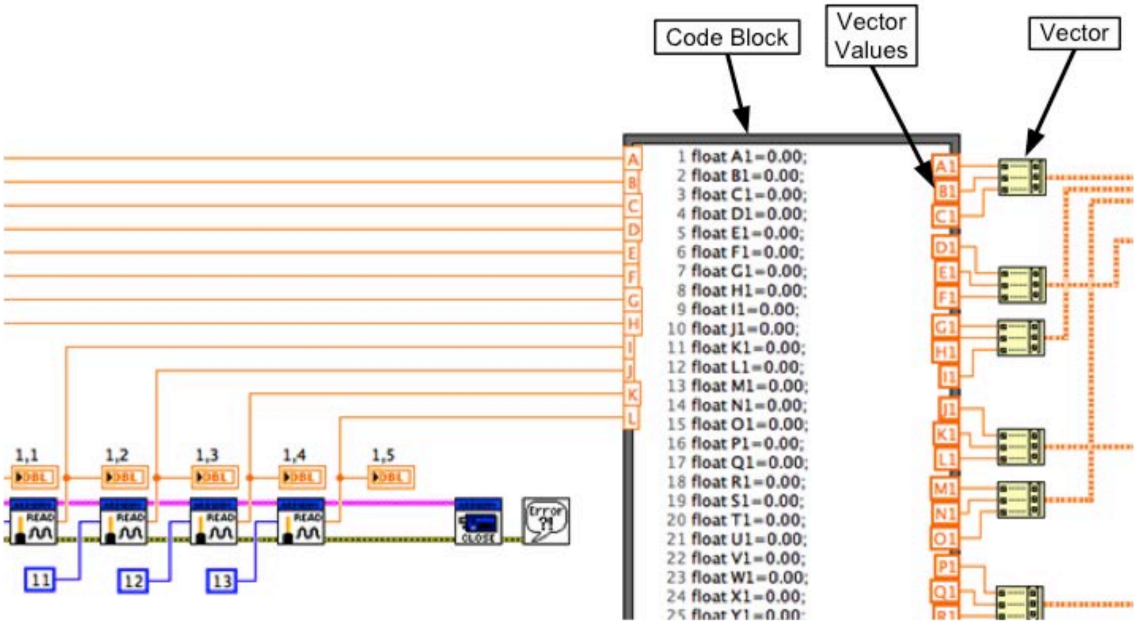


Figure 5.9 Data Input

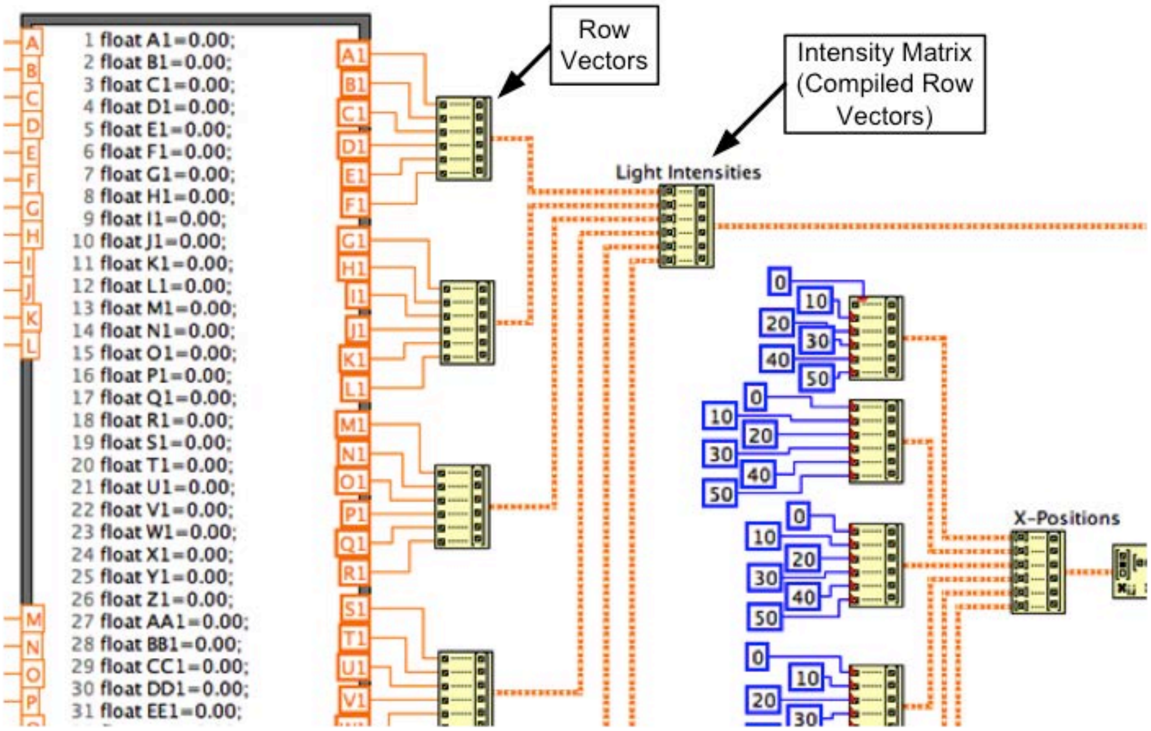


```
float A1=0.00;
float B1=0.00;
float C1=0.00;
float D1=0.00;
float E1=0.00;
float F1=0.00;
float G1=0.00;
float H1=0.00;
float I1=0.00;
float J1=0.00;
float K1=0.00;
float L1=0.00;
float M1=0.00;
float N1=0.00;
float O1=0.00;
float P1=0.00;
float Q1=0.00;
float R1=0.00;
float S1=0.00;
float T1=0.00;
float U1=0.00;
float V1=0.00;
float W1=0.00;
float X1=0.00;
float Y1=0.00;
```

Figure 5.10 Data Filtering

The code block in Figure 5.10 serves one purpose. Using a constant factor, it filters out signals coming from sensors that are only detecting ambient light, allowing the calculations to utilize only the sensors detecting light from the laser.

After passing through the filter, the resulting values are arranged into vectors that represent the rows of the Grid Array. These are then arranged into a matrix as shown in the area labeled "Intensity Matrix" in Figure 5.11.



```
float A1=0.00;
float B1=0.00;
float C1=0.00;
float D1=0.00;
float E1=0.00;
float F1=0.00;
float G1=0.00;
float H1=0.00;
float I1=0.00;
float J1=0.00;
float K1=0.00;
float L1=0.00;
float M1=0.00;
float N1=0.00;
float O1=0.00;
float P1=0.00;
float Q1=0.00;
float R1=0.00;
float S1=0.00;
float T1=0.00;
float U1=0.00;
float V1=0.00;
float W1=0.00;
float X1=0.00;
float Y1=0.00;
float Z1=0.00;
float AA1=0.00;
float BB1=0.00;
float CC1=0.00;
float DD1=0.00;
float EE1=0.00;
```

Figure 5.11 Data Sorting

The X and Y-coordinates of each photodiode are placed into one of two matrices like that shown in the area highlighted with a box in Figure 5.12. One matrix contains the X-coordinate of each photodiode while the other matrix contains the Y-coordinate of each photodiode. Each element of the light-intensity matrix (described above and shown in Figure 5.11) is multiplied by each element of the X-matrix and then divided by the sum of each element of the intensity matrix to produce the X-coordinate values. The same is done with the Y-matrix to produce the Y-coordinate value. The calculations are based on equations (5.3) and (5.4).

$$X = \frac{x_1 I_1 + x_2 I_2 + \cdots + x_n I_n}{I} \qquad (5.3)^*$$

$$Y = \frac{y_1 I_1 + y_2 I_2 + \cdots + y_n I_n}{I} \qquad (5.4)^*$$

*Same as equations shown earlier.

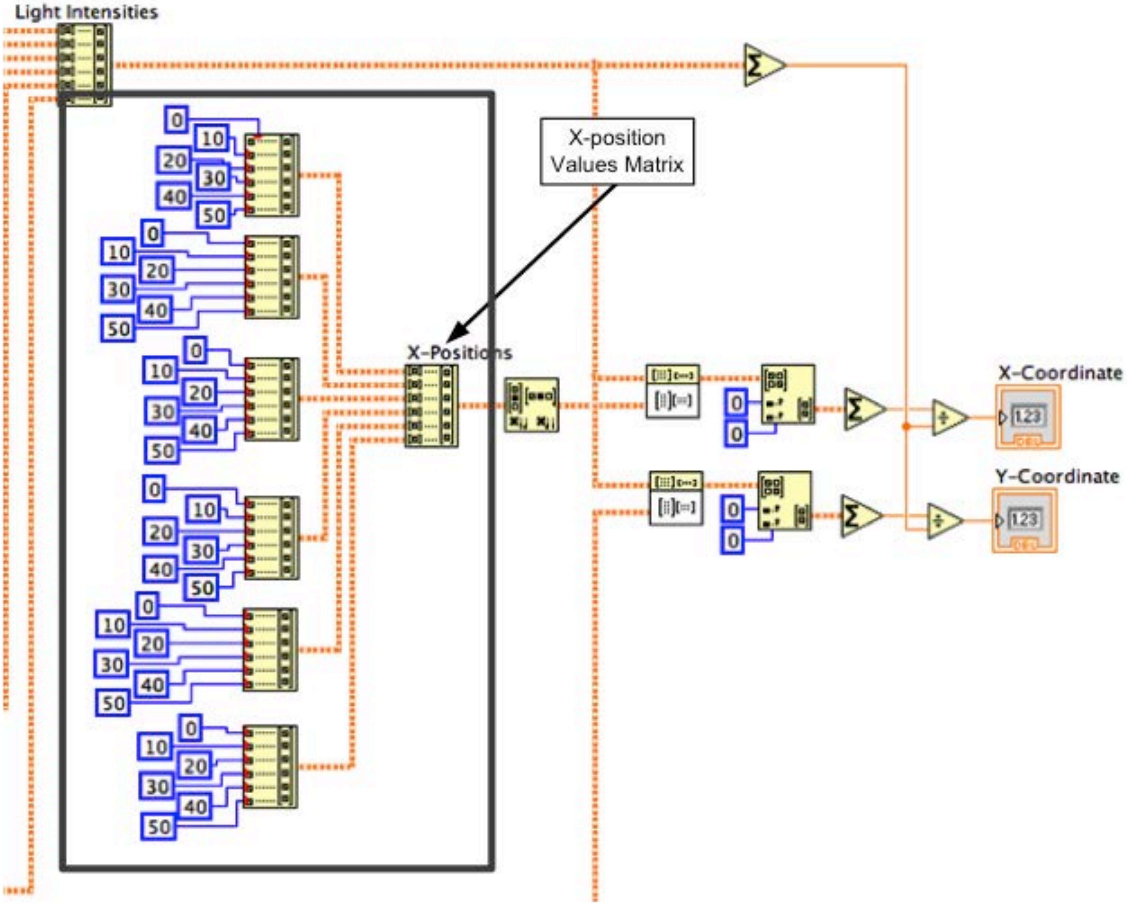
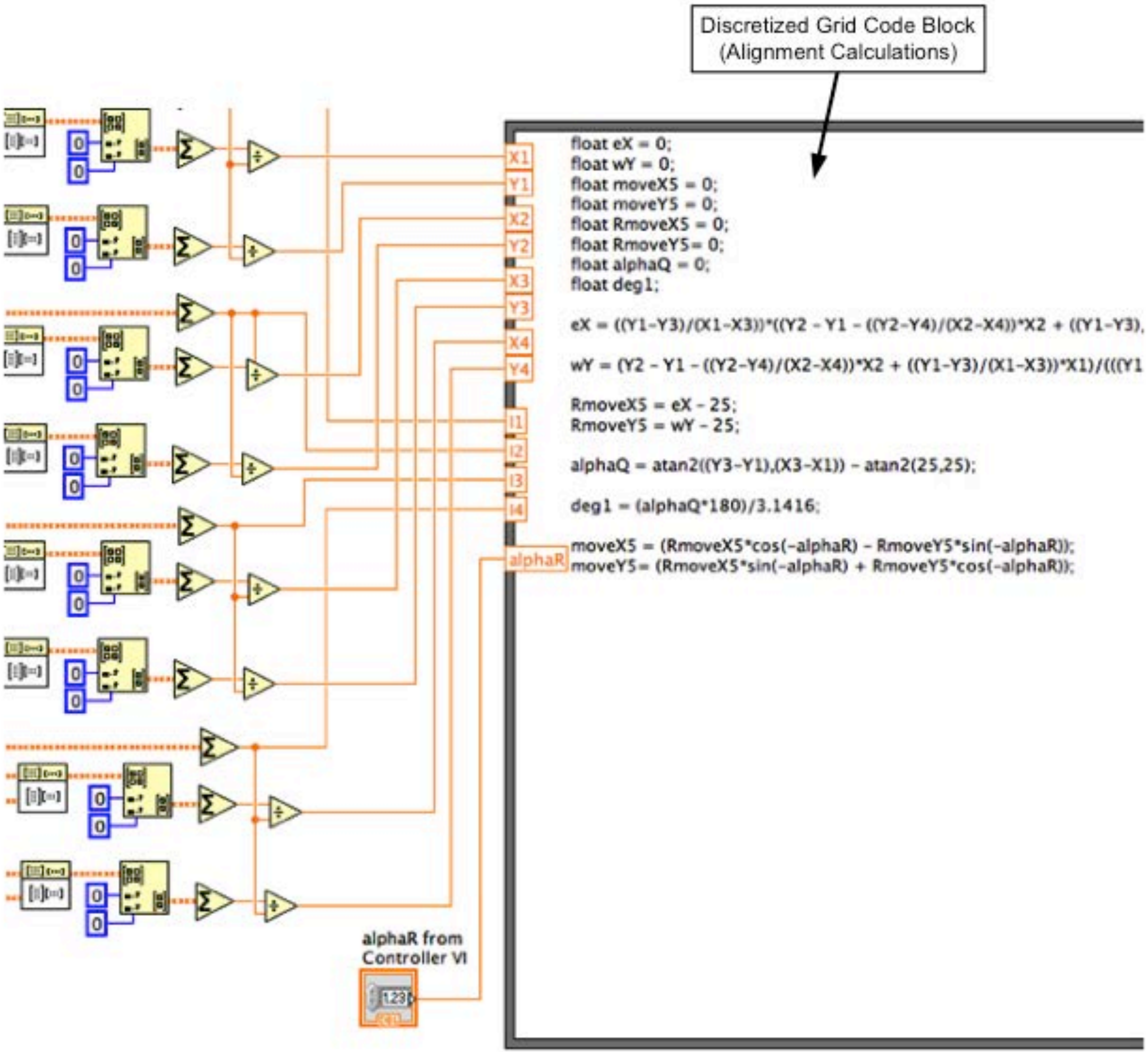


Figure 5.12 Creation of X and Y-coordinate Values

The first part of the program that discretizes the grid into four quadrants functions identically to the program that utilizes the full grid all at once. The software components that read the data from the DAQ and the filter are the same. After passing through the filter, the light intensity values are broken up into four different matrices, each one representing a quadrant of the Grid Array. The same mathematical processes given by equations (5.3) and (5.4) are used for each quadrant. The X and Y-coordinates of the laser in each quadrant are input into another code block as shown in Figure 5.13.



```
float eX = 0;
float wY = 0;
float moveX5 = 0;
float moveY5 = 0;
float RmoveX5 = 0;
float RmoveY5= 0;
float alphaQ = 0;
float deg1;

eX = ((Y1-Y3)/(X1-X3))*((Y2 - Y1 - ((Y2-Y4)/(X2-X4))*X2 + ((Y1-Y3),

wY = (Y2 - Y1 - ((Y2-Y4)/(X2-X4))*X2 + ((Y1-Y3)/(X1-X3))*X1)/(((Y1

RmoveX5 = eX - 25;
RmoveY5 = wY - 25;

alphaQ = atan2((Y3-Y1),(X3-X1)) - atan2(25,25);

deg1 = (alphaQ*180)/3.1416;

moveX5 = (RmoveX5*cos(-alphaR) - RmoveY5*sin(-alphaR));
moveY5= (RmoveX5*sin(-alphaR) + RmoveY5*cos(-alphaR));
```

Figure 5.13 Quadrant Coordinates Input and Rotation Command Calculation

The code block uses the coordinate inputs to create the rotation and translation commands to send the robot for use in the second step of the alignment. It has six cases to choose from. The individual diagrams in Figure 5.15 are pictorial representations of each case. **Case 1** represents a small counterclockwise rotational misalignment of the plate about the lead laser. **Case 2** represents a small clockwise rotational misalignment. The remaining four cases differ in that they have larger rotational misalignments, meaning that one to two lasers are out of the sensing region. When this happens, the system analyzes the distance between the remaining lasers in the sensing region to determine the direction of rotation. For example, in **Case 3** the fact that the upper right quadrant does not detect anything is a good indication that there is a large clockwise rotation. To verify this, the distance between the lead laser and the lower right laser is the misaligned condtion is compared against the distance between the lead laser and the upper right laser in the ideal condition to see if they are the same distance apart. Additionally, it is also checked that the distance between the lead laser and the laser in the lower left quadrant are the distance that the lead laser should be from the lower right laser (this should be the furthest distance between any of the lasers).

The distance between the lead laser and the upper right laser is $\sqrt{(10mm)^2 + (40mm)^2} = 41.23mm$ (see Figure 5.14 below). This is also the distance between the lead laser and the lower left laser. The distance between the lead laser and the lower right laser is $\sqrt{(40mm)^2 + (40mm)^2} = 56.57mm.$ Therefore, in Case 3, the lower right quadrant should read a point that is approximately 41.23mm away from the lead laser and the lower left quadrant should read a point that is approximately 56.57mm away from the lead laser. Measurements and equations for getting the measurements calculated above are shown below in Figure 5.14.

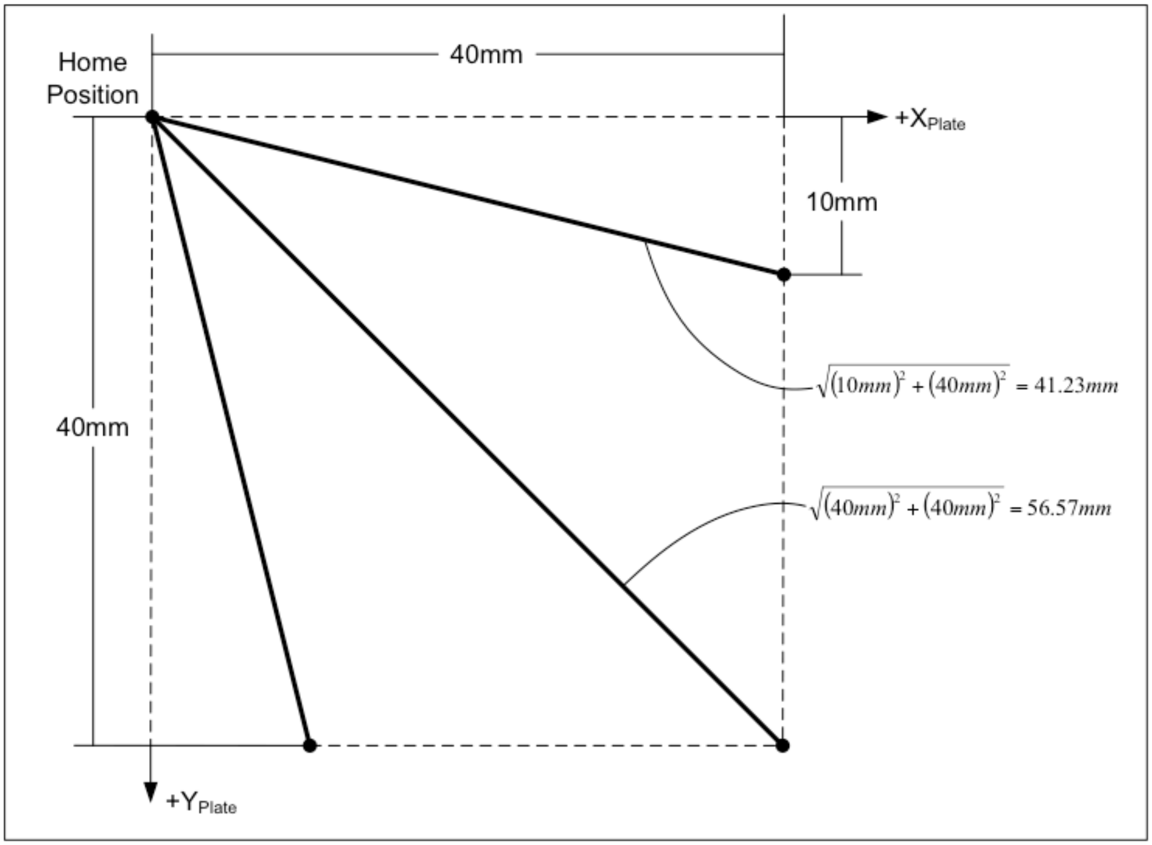

Home Position
40mm
+X_Plate
10mm
$\sqrt{(10mm)^2 + (40mm)^2} = 41.23mm$
40mm
$\sqrt{(40mm)^2 + (40mm)^2} = 56.57mm$
+Y_Plate

Figure 5.14 Laser Spacing

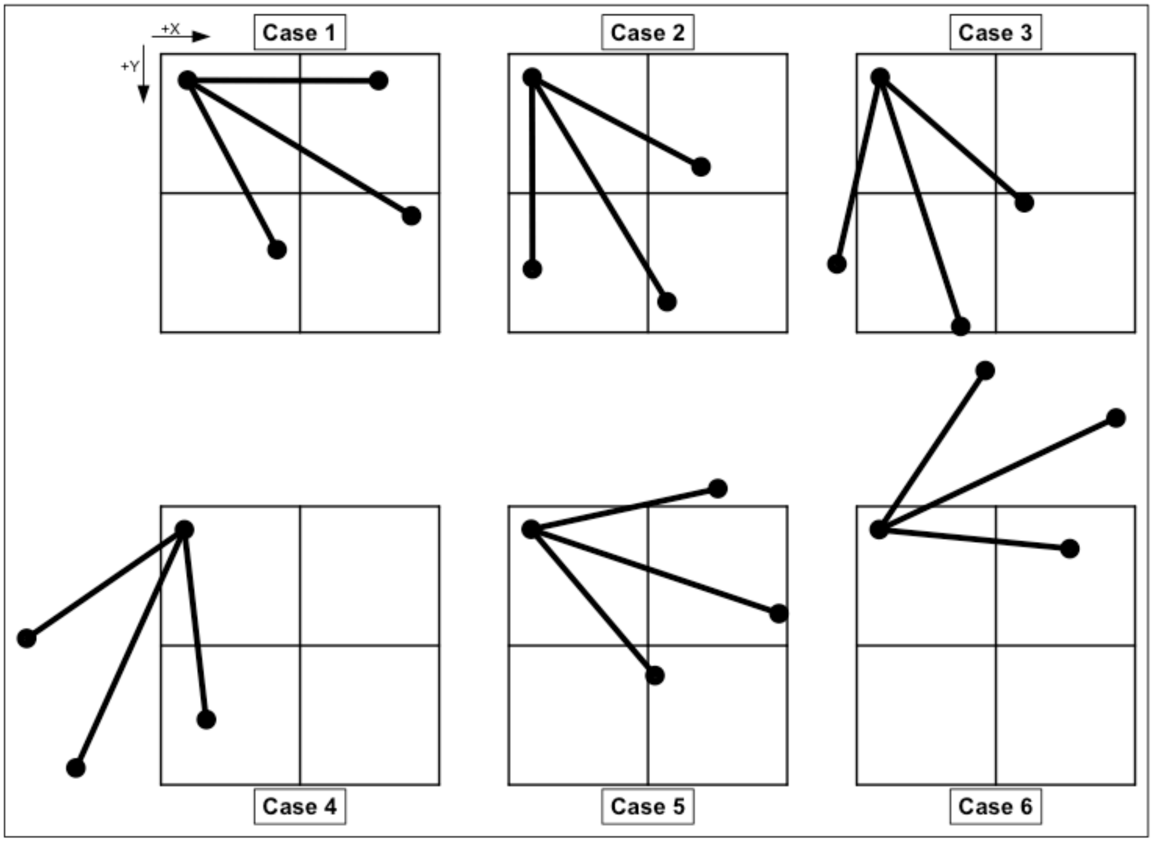
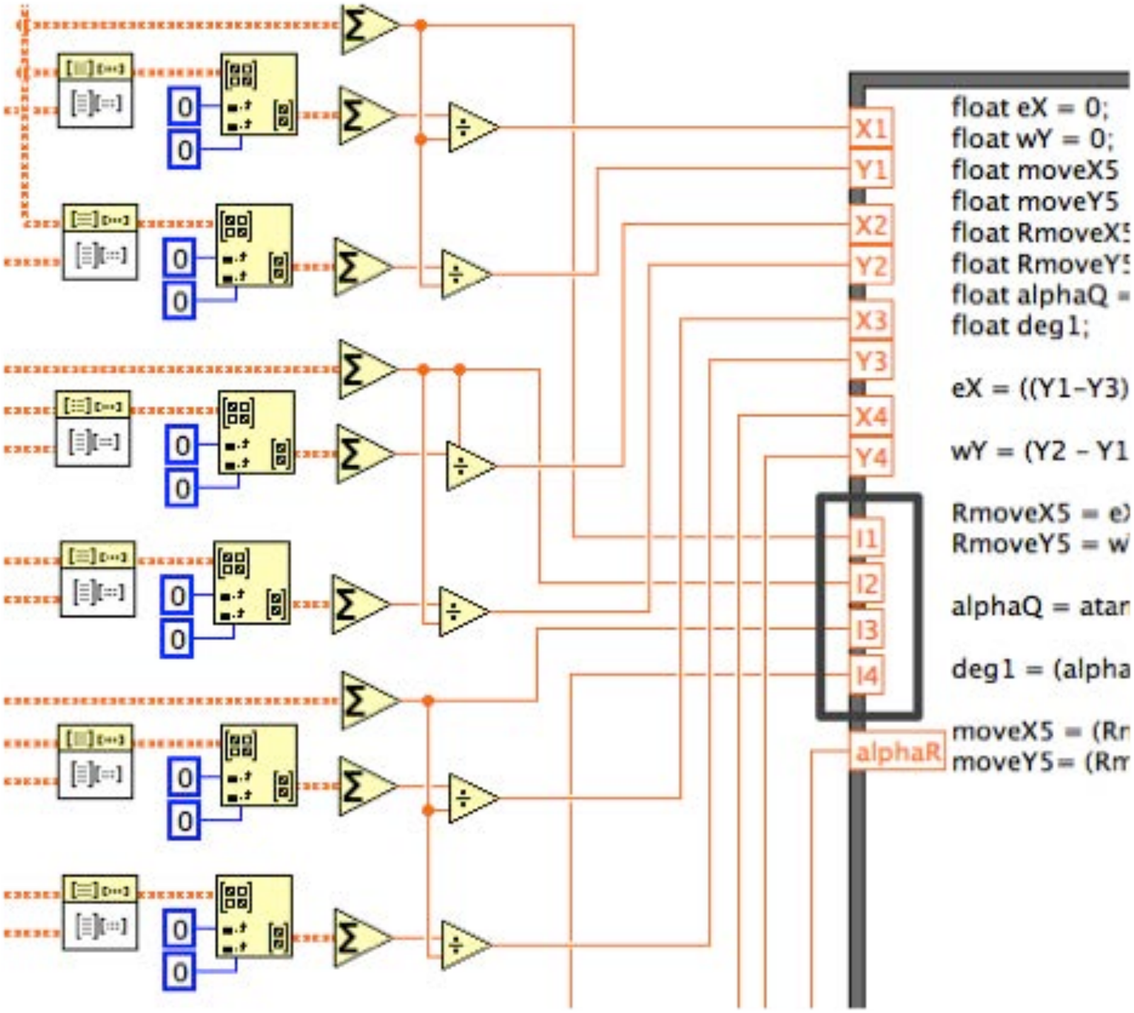


Figure 5.15 Rotation Cases of Second Alignment Step

Ideally the system would use two lasers to determine the initial orientation of the laser plate relative to the Grid Array. For **Case 1**, **Case 2**, **Case 5**, and **Case 6** the laser positions in Quadrant 1 and Quadrant 2 will be used to calculate the rotational orientation of the laser plate. In **Case 3** and **Case 4** the laser locations in Quadrant 1 and Quadrant 4 will be used to calculate orientation. Because of error in the position detection, which will be discussed later, measuring the orientation using only two points causes the system to behave sporadically. Therefore, another method is used for aligning, but it only works when all four lasers are on the grid, meaning that it only works for **Case 1** and **Case 2**. This new method uses diagonal lasers and finds the equation of the line that connects both points. The intersecting point of the lines between the two sets of diagonal points is used as the center of the laser plate. This method has proven to be more reliable.

In the first method, it is necessary to determine if a quadrant has a laser in it in order to select which case from above to use. To determine that a quadrant is not detecting a laser, the summed intensity of each quadrant is fed into the code block as shown in the outlined area on the right side of Figure 5.16. If the intensity is equal to zero, then there is no laser being detected in that quadrant.



```
float eX = 0;
float wY = 0;
float moveX5
float moveY5
float RmoveX5
float RmoveY5
float alphaQ =
float deg1;

eX = ((Y1-Y3)

wY = (Y2 - Y1

RmoveX5 = e
RmoveY5 = w

alphaQ = atar

deg1 = (alpha

moveX5 = (Rr
moveY5= (Rm
```

Figure 5.16 Intensity in each Quadrant

As a result of the case selection in the code block and the subsequent calculations in the first method or the calculations in he second method, the code block outputs the positional value of each laser diode, as well as the translation and rotation needed to align the center of the laser module to the center of the Grid Array. The boxed variables on the left half of Figure 5.17 labeled **moveX5**, **moveY5**, and **deg1** are the translation in the X-direction, the translation in the Y-direction, and the rotation needed to align the center of the laser plate to the grid.

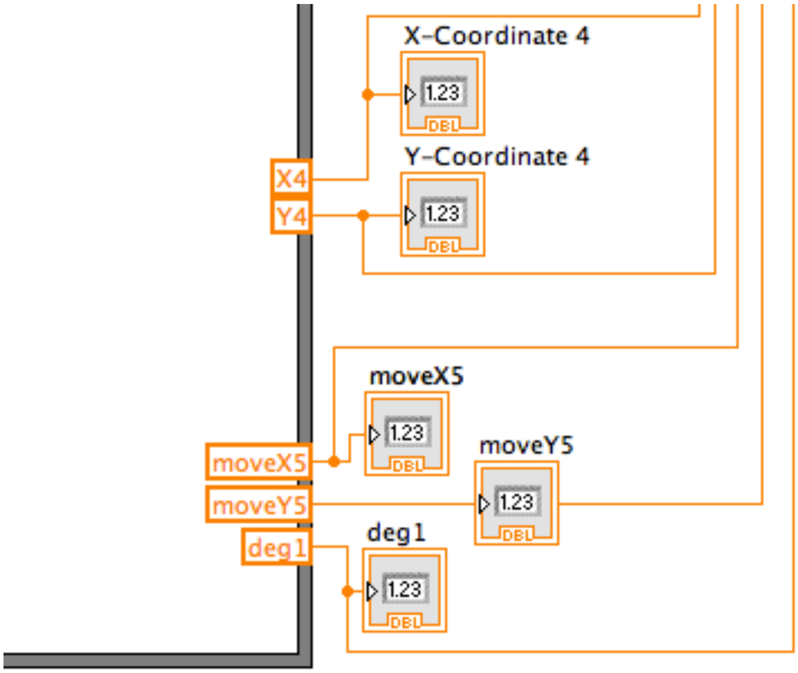


Figure 5.17 Second Step Translation and Rotation Outputs

The higher-level control program (or VI) brings together the measurements taken by both the full-grid and quadranted-grid VI's and feeds the values into movement functions for the robot. When the program is run, the first step is to initialize the robot as shown in Figure 5.18. This step opens the communication port with the robot, determines the order to initialize the axes of the robot, and determines if the robot should be homed or not. The following step shown in Figure 5.19 is to move the gripper 70mm in the Z-direction. This clears the gripper from the base of the robot, allowing it a larger movement area.

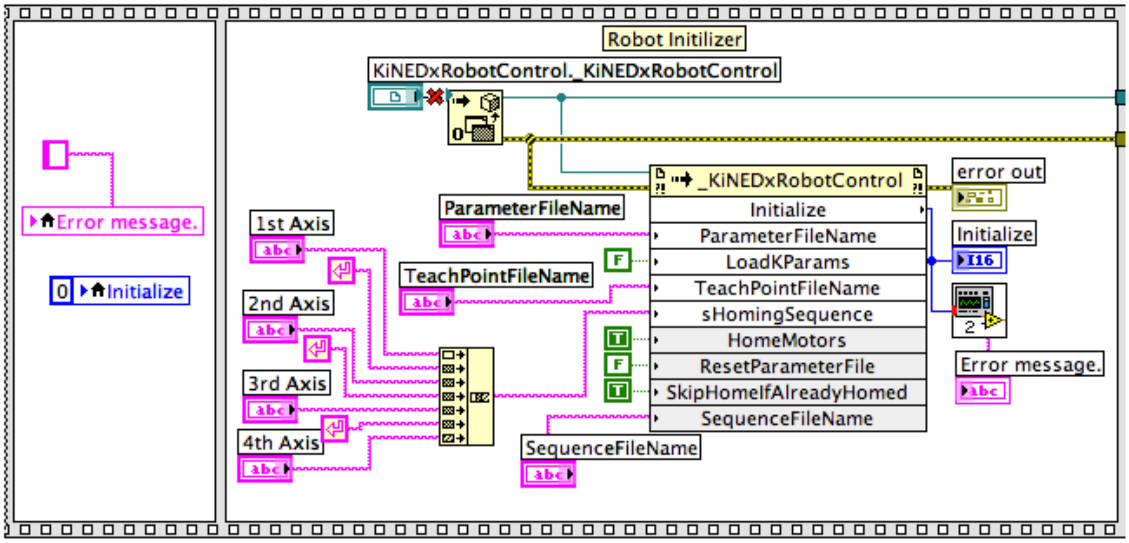


Figure 5.18 Program Initialize

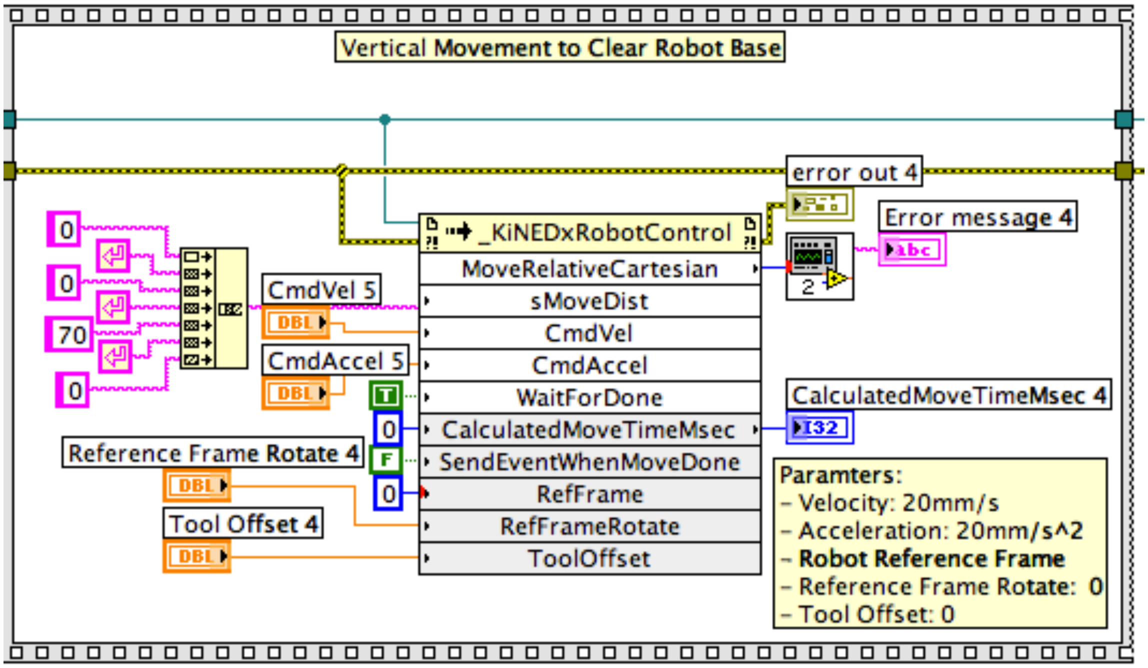


Figure 5.19 Gripper Clearance

The program then enters the while loop that contains the alignment functionality of the program. Within the while loop resides a case block. When the case is set to 0, the program does nothing, but once the alignment button on the GUI is pressed, **Case 1** of the case block is run, which performs the alignment. The **Start Primary Alignment** block indicated with a label in Figure 5.20 is the start button for the alignment. It enters into **Case 1** and the first step is to move the robot 10mm along the positive X-axis in the robot coordinate frame. After the robot has performed the move initiated by the function in Figure 5.20, the program waits 300ms and then runs the full-grid VI to get the X and Y-position values for the lead laser. The component of the program associated with this is shown in Figure 5.21.

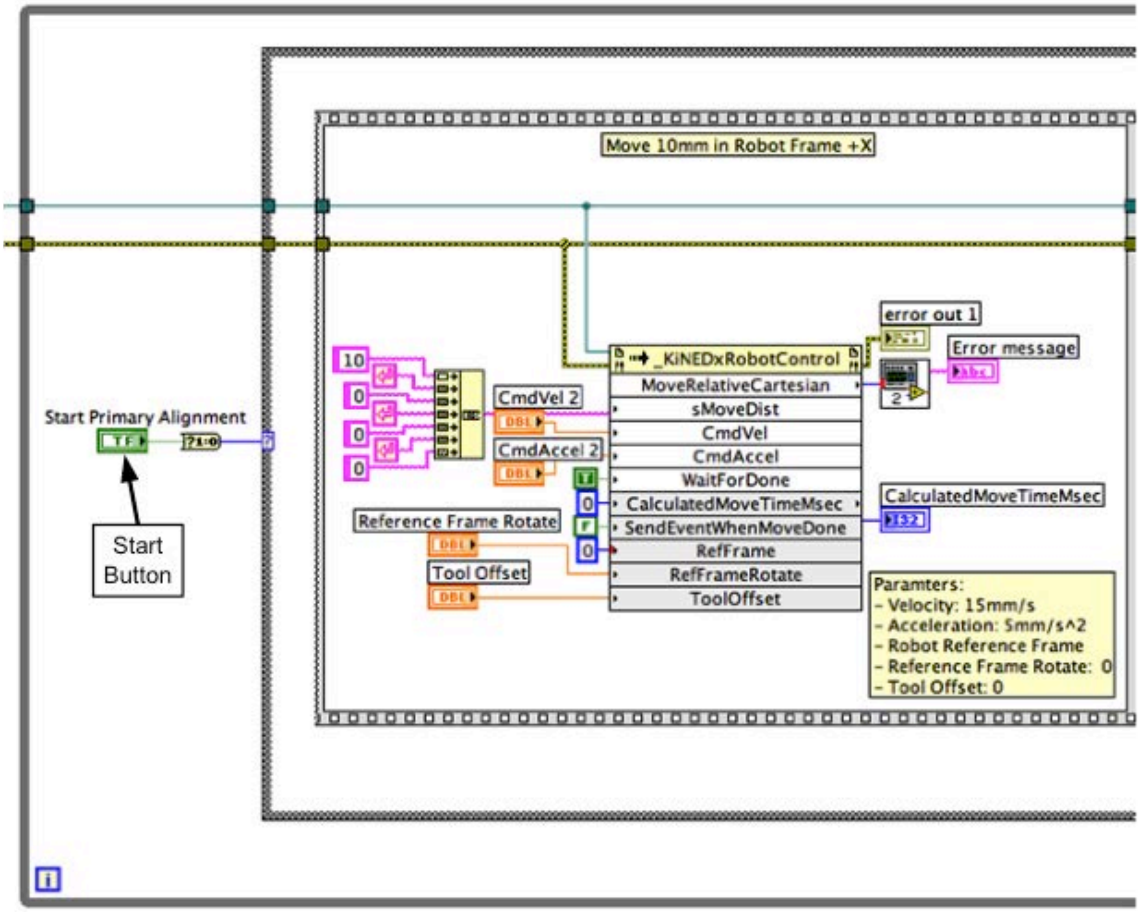


Figure 5.20 Case Selection and First Alignment Movement

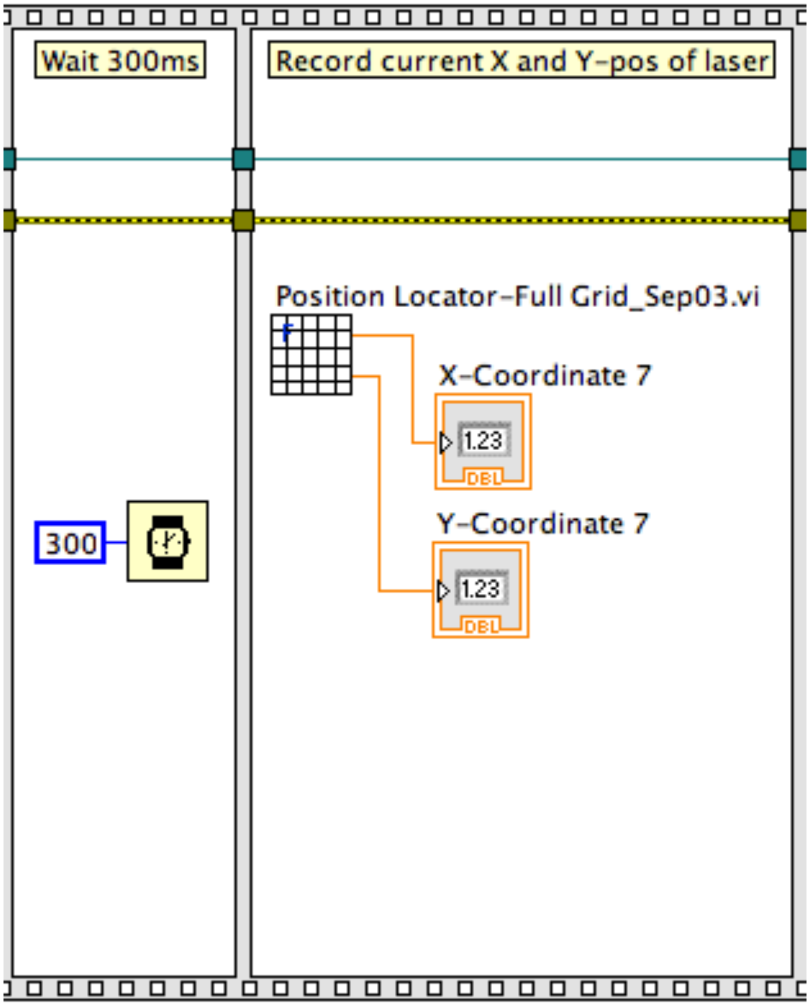
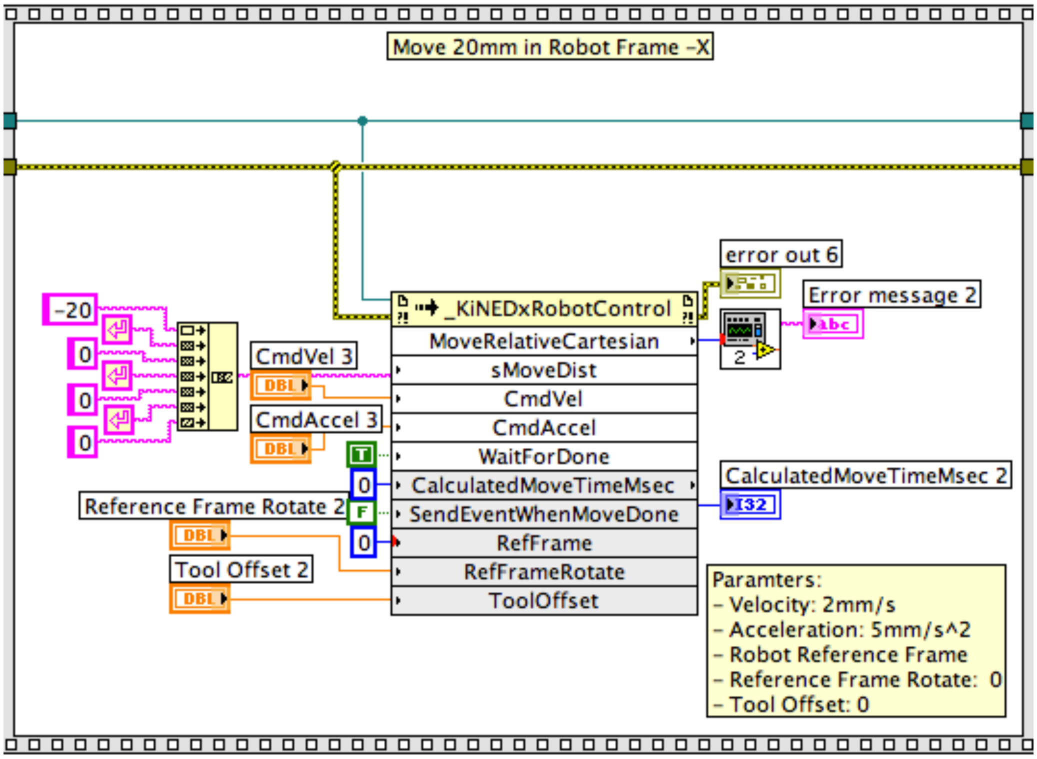

Wait 300ms
Record current X and Y-pos of laser
Position Locator-Full Grid_Sep03.vi
X-Coordinate 7
1.23
DBL
Y-Coordinate 7
1.23
DBL
300

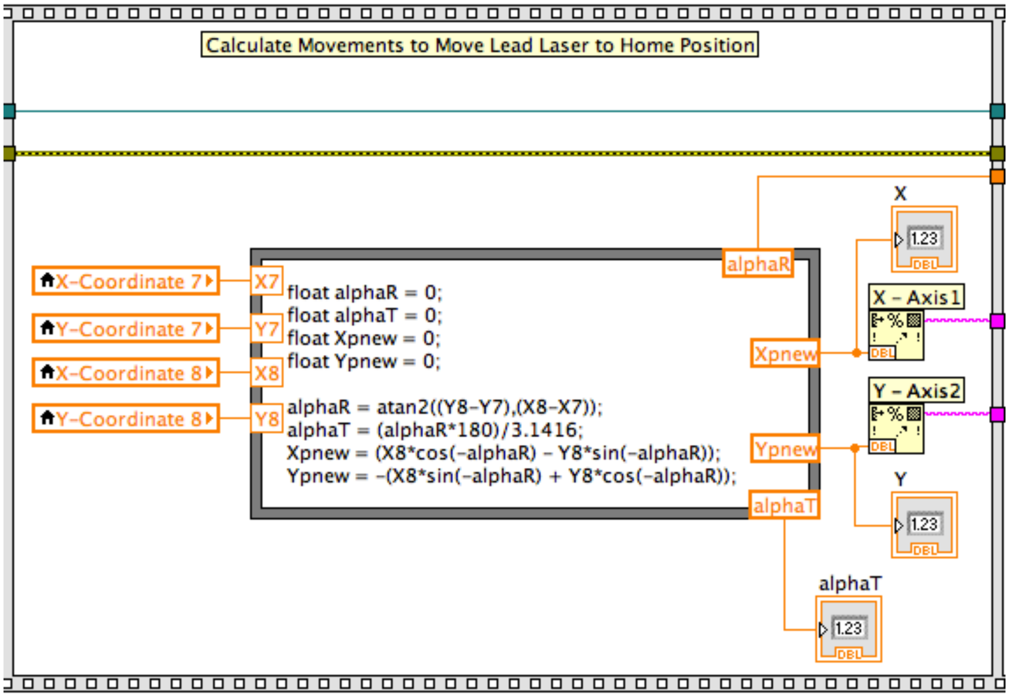
Figure 5.21 First Position Detection with Full-Grid

The following step, shown in Figure 5.22, is similar to that shown in Figure 5.20, except that the robot movement is performed along the negative X-axis of the robot frame of reference. The program performs an identical step to that shown in Figure 5.21 and collects the coordinates of the second location of the lead laser. Using these two points, it performs a calculation step as shown in Figure 5.23 below. Equations (3.1) - (3.3) are used to convert the distance needed to move the lead laser to the Home Position from the grid coordinate frame to the robot coordinate frame. When the values are output from the code block, each value is both displayed on screen and converted to the appropriate data type to be read in by the following function. This function in Figure 5.24 receives the commands from the code block and then moves the robot in the robot coordinate frame accordingly to align the lead laser with the Home Position.



Paramters:
- Velocity: 2mm/s
- Acceleration: 5mm/s^2
- Robot Reference Frame
- Reference Frame Rotate: 0
- Tool Offset: 0

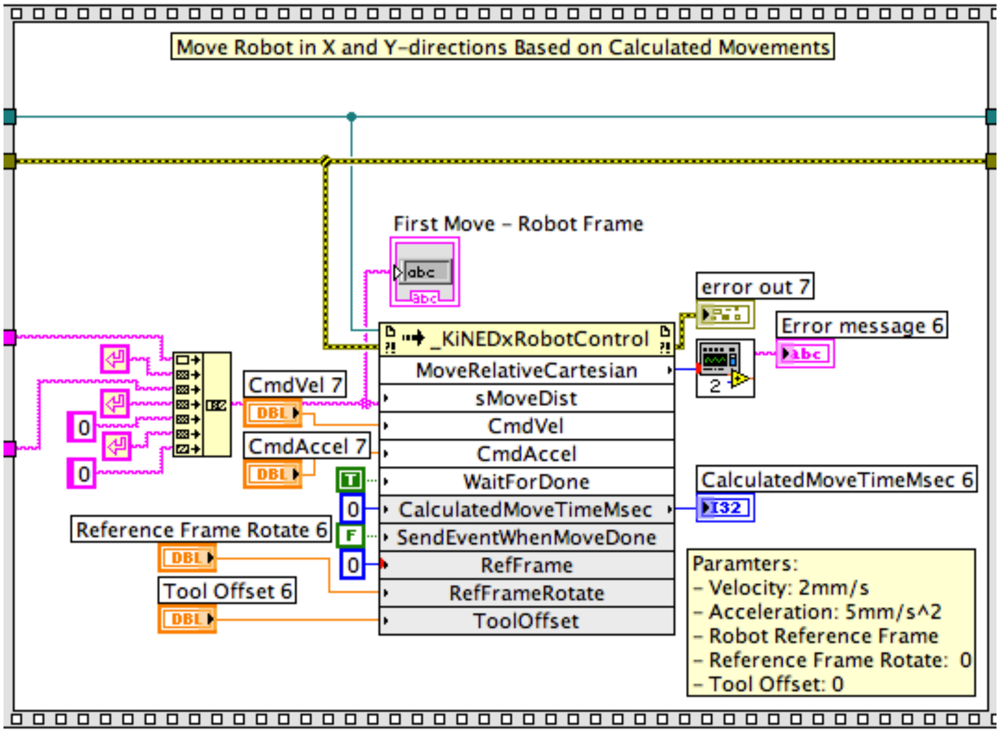
Figure 5.22 Second Alignment Movement



```
float alphaR = 0;
float alphaT = 0;
float Xpnew = 0;
float Ypnew = 0;

alphaR = atan2((Y8-Y7),(X8-X7));
alphaT = (alphaR*180)/3.1416;
Xpnew = (X8*cos(-alphaR) - Y8*sin(-alphaR));
Ypnew = -(X8*sin(-alphaR) + Y8*cos(-alphaR));
```

Figure 5.23 Vector Calculation Step



Paramters:
- Velocity: 2mm/s
- Acceleration: 5mm/s^2
- Robot Reference Frame
- Reference Frame Rotate: 0
- Tool Offset: 0

Figure 5.24 Resulting X and Y-Translations

After a further time delay to allow the system to move the robot, an inset while loop functioning as a feedback loop is started to measure the position of the robot and move until the center of the laser plate is less than or equal to 1.5mm in both the X and Y-directions away from the center of the Grid Array. The loop is shown broken into halves in Figure 5.25 below. After entering the loop another laser position reading is taken. This time, it uses the quadrant-grid VI and gets as a result the X and Y-movements and the rotation needed to align the center of the laser plate with the center of the Grid Array. This step is shown in Figure 5.26 below. Based on these values, the robot gripper is then rotated relative to the tool reference frame and translated relative to the robot reference frame. The use of the tool reference frame can be seen by the value of 1 in the **RefFrame** field in Figures 5.27, and the value of 0, denoting the robot reference frame, in the **RefFrame** field is seen in Figure 5.28 in the seventh input line under the function named **MoveRelativeCartesian**. The function in Figure 5.27 performs the rotation while the function in Figure 5.28 performs the two translations.

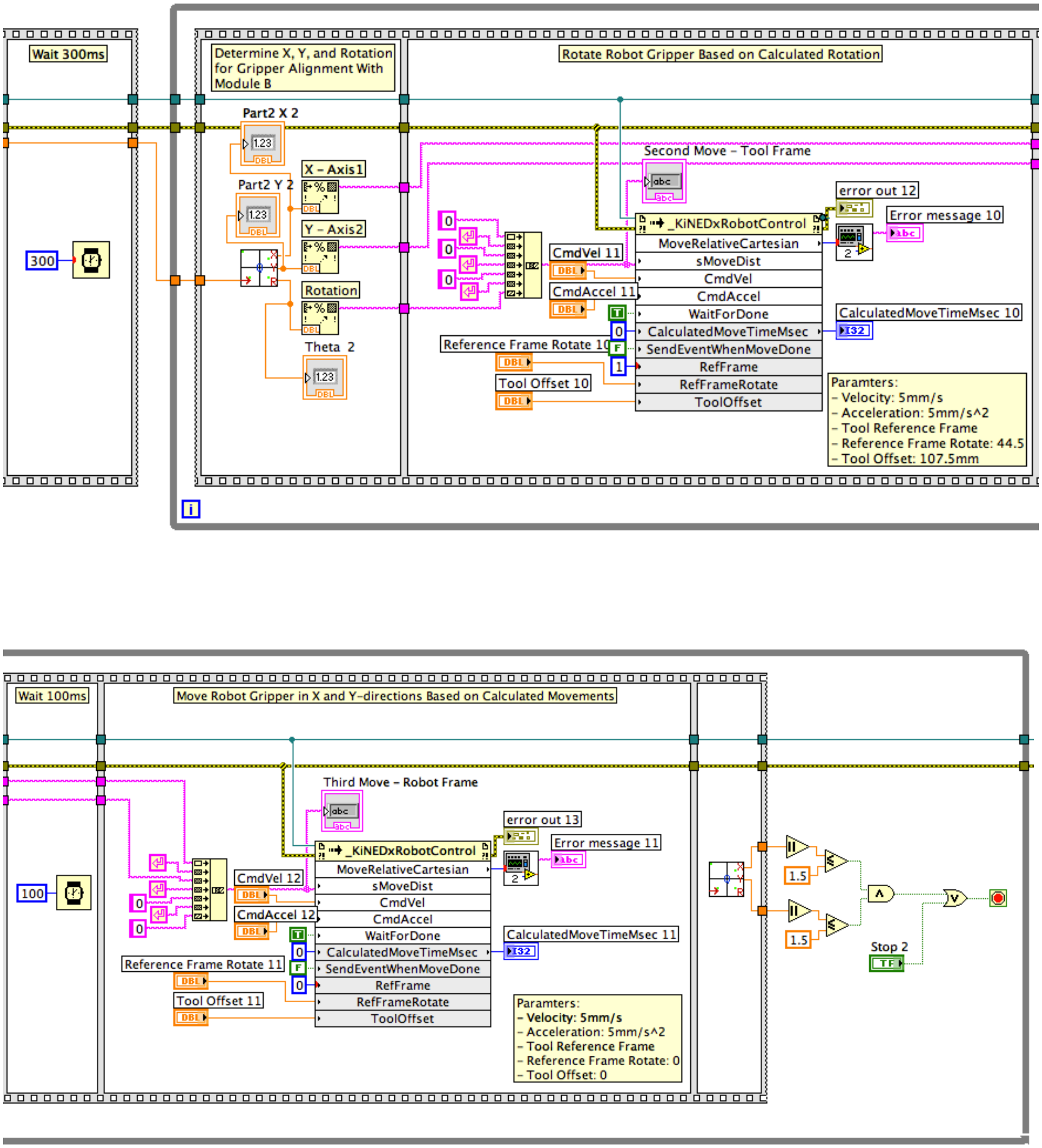


Figure 5.25 Feedback While Loop

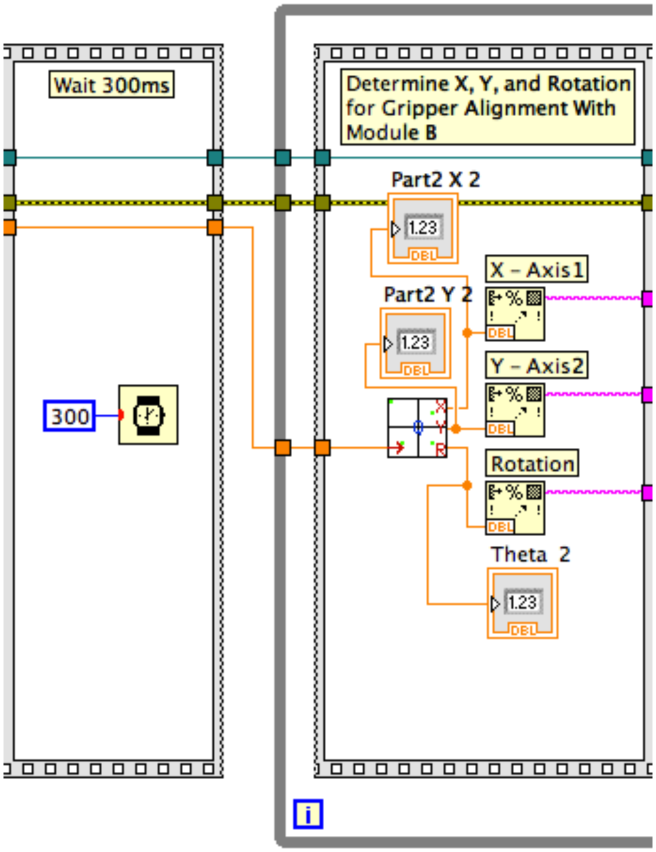
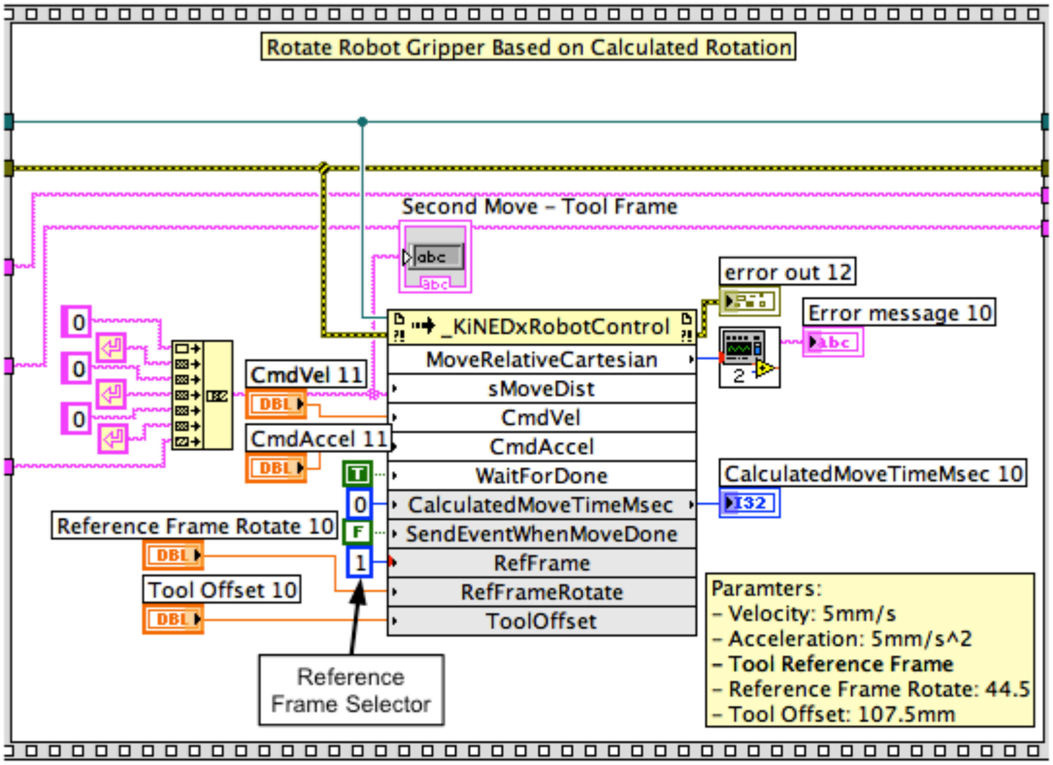
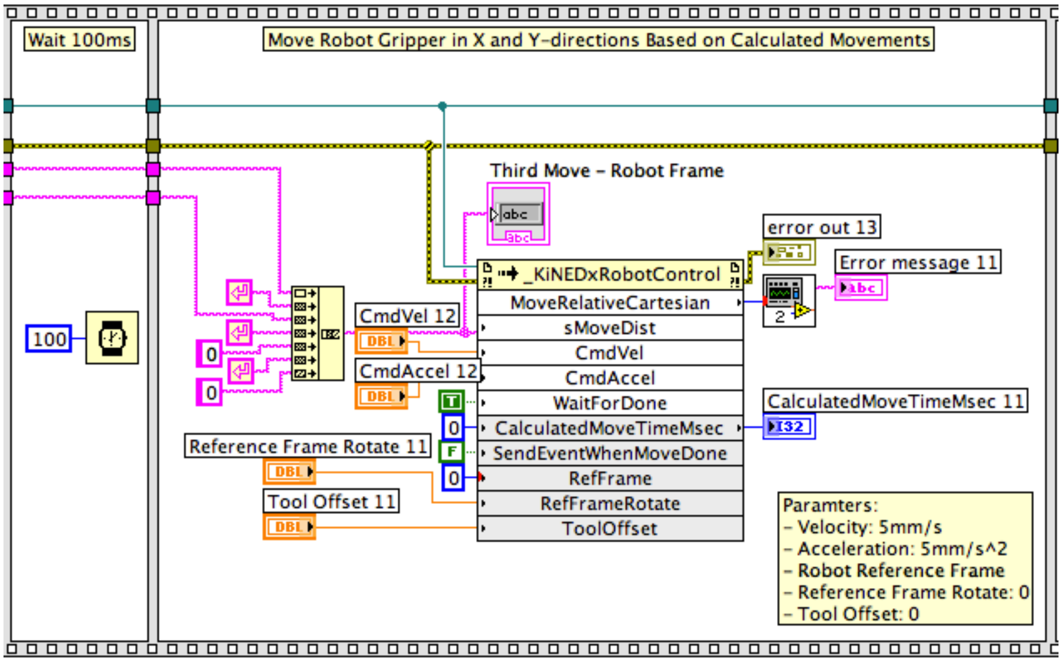


Figure 5.26 Time Delay and Calling the Quadrant-Grid VI to Find the X, Y, and Rotation Needed to Align the Centers of the Modules



Paramters:
- Velocity: 5mm/s
- Acceleration: 5mm/s^2
- Tool Reference Frame
- Reference Frame Rotate: 44.5
- Tool Offset: 107.5mm

Figure 5.27 Rotation Based on Quadrant-Grid VI Output



Paramters:
- Velocity: 5mm/s
- Acceleration: 5mm/s^2
- Robot Reference Frame
- Reference Frame Rotate: 0
- Tool Offset: 0

Figure 5.28 X and Y-Translation Based on Quadrant-Grid VI Output

After adjusting the rotation of the laser plate relative to the Grid Array and updating the translation, another position measurement is taken. This compares a calculated center of the laser plate to the center of the Grid Array. The difference in the X and Y-directions is compared as shown in Figure 5.29 below. If the difference between the two plate centers is less than 1.5mm in both the X and Y-directions, then the system will stop the loop and will therefore be complete with the alignment. If instead the centers are not within the error allowance range, then the system will start the loop over again, read the distance between the centers, and then rotate and translate the robot gripper accordingly. This will happen until the laser plate center is within the allowable error range of the center of the Grid Array. This loop is a feedback loop to improve the alignment between the laser plate and the Grid Array. An additional stop button is included at the end of the while loop to allow an operator to stop the feedback loop in the unlikely event that the distance between the centers diverge and the error grows increasingly large.

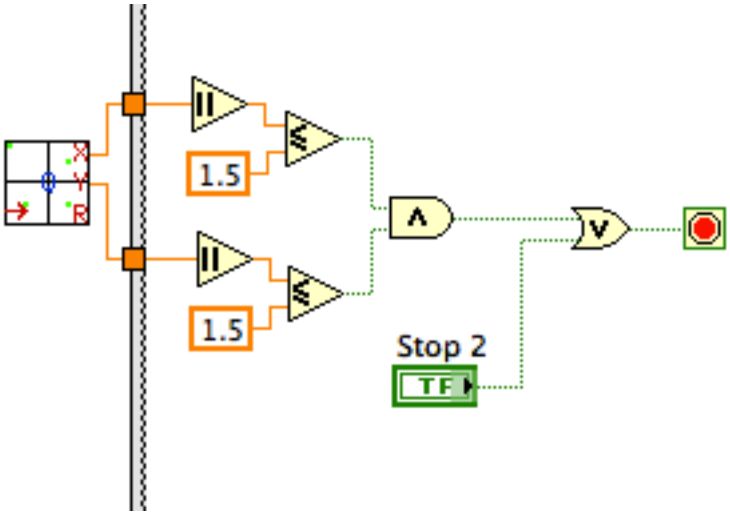


Figure 5.29 Laser Plate to Grid Center Error Allowance Comparison

After all of the alignment has been completed and the program Stop button has been pushed, as represented by the box labeled **Stop** in the lower left half of Figure 5.30, the outer while loop is exited. The program then runs a function labeled **ShutDown**, displayed on the right side of Figure 5.30, which closes the COM port with the robot, ending the main control program of the alignment system.

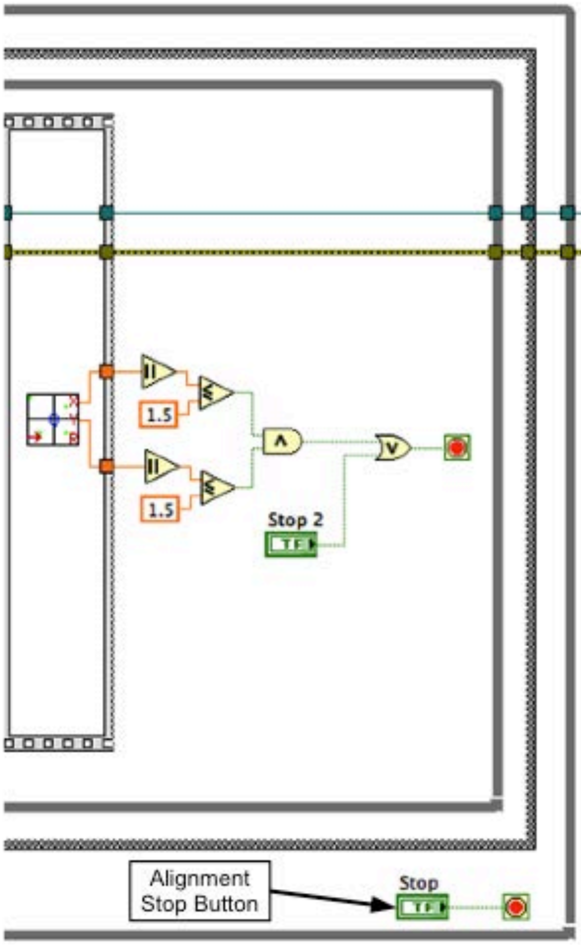


Figure 5.30 End of Program and Robot Communications Shutdown

Three programs are used to form the control system for the alignment process. The first two use the grid array to detect the position of either one laser or four lasers. The measurement values found using these programs are used in the main control program, which calculates the movements needed to align the laser plate to the Grid array and then outputs the commands to the robot. The final alignment step is imbedded in a feedback loop that ensures that the laser plate alignment to the Grid Array is within a specified error range.

6. SYSTEM INITIALIZATION

6.1 Tool Coordinate System Setup

To use a tool coordinate system, it is necessary to know the center of Module A (laser module). To achieve this, a laser is placed in the center of the plate, which also corresponds to the center of the laser square configuration formed by the three secondary lasers. The approximate distance between the robot wrist joint and the center of the plate is then measured. Using this value, a rotation is made along with adjustments to the offset value until the appropriate value is achieved. Figure 6.1 shows the location of the center alignment laser on Module A. For a SCARA robot, such as that used in the experimentation, the robot actually moves in a tear-drop motion, but the key to the alignment is that the laser point is at the same spot before and after the rotation.

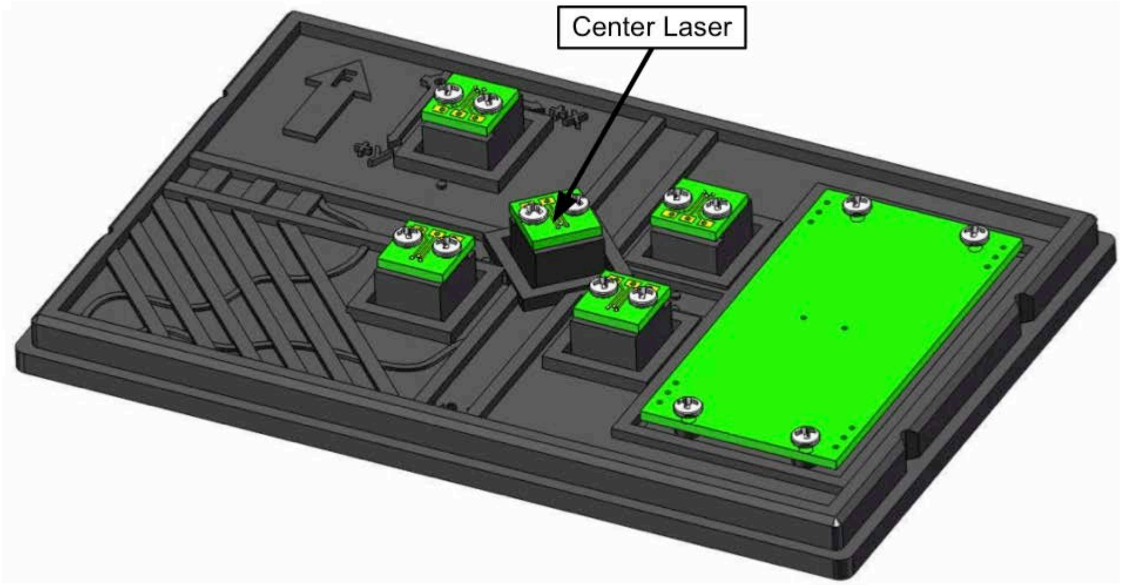


Figure 6.1 Center Alignment Laser

6.2 Light Intensity Threshold Selection

The photodiodes detect ambient light from various sources in the room. Therefore, it is necessary to operate the system in low-light situations. Despite the low light, any form of light other than the laser detected by the photodiodes will be included in the center-of-gravity calculation and affect the position calculation. To eliminate this effect, a software filter is used to remove the signal input from photodiodes receiving ambient light. Because the light intensity will be small compared to the laser output, any photodiode value below a threshold value is set to zero. In order to get the correct value, it is essential to measure the ambient light intensity before running any alignment. This is performed by removing any lasers from above the Grid Array. The position detection algorithm is run so that system will detect the output reading of each photodiode. A value above the maximum output value is then set as the threshold. If this value is set correctly, the position detection algorithm will give a divide by zero error because the intensity value of each photodiode used in the calculations will be set to zero before being used in the calculation. To ensure the best results, it is essential to reset the threshold value whenever there are any environmental lighting changes.

The maximum photodiode voltage output is approximately 4.5mm, which is achieved when a laser spot is centered over a photodiode. Additionally, most threshold values are within 0.03V and 0.1V, meaning that they are quite small compared to the values generated by photodiodes being struck by laser light. Therefore, the likelihood of loosing valuable data is small, and any lost data would have a small effect on the calculation.

7. IMPLEMENTATION AND RESULTS

The best way to test the system is to run it through the full alignment process. The first part of this chapter covers the accuracy of measuring individual points, which is important in the system's ability to measure an accurate robot X-vector. The second part includes the first step of calculating the X-vector of the robot relative to the X-vector of the Grid Array and then the result of running the second step of the alignment starting from two different orientations. Each of these orientations ran 30 trials and the results were plotted. The difference between the un-aligned case and the aligned case is compared, as well as the difference between what the system measured and what was observed. The error measures were calculated based on the difference between system-measured and observed values.

7.1 Test 1

To verify the functionality of the sensors and the validity of the control algorithm, two types of tests were run. The first was to move a single laser methodically across the Grid Array and compare the algorithm output of the position to that measured by hand. Measurements were taken at 5mm increments in both the X and Y-directions. Although the grid has a 10mm by 10mm spacing between sensor centers, measurements were taken in a 5mm by 5mm grid to measure the accuracy of the system in detecting the position of a laser point between sensors. Although the measurements are within ±0.5mm taken at a resolution of 0.5mm, they help provide a picture of the accuracy of the system.

The resolution of the algorithm output is 0.1mm. The result of the test shows an absolute average error of 0.5mm in the X-direction and 0.3mm in the Y-direction. The maximum absolute error is 2.9mm in the X-direction and 2mm in the Y-direction. Despite the large absolute error, the mode of the error is 0.0mm in the X-direction and 0.1mm in the Y-direction. Figure 7.1 is a frequency plot displaying the frequency of error along the X-axis in specific ranges. It shows that out of the 121 points taken along the grid, 79 had an error of 0.5mm or less and 25 had an error between 0.6mm and 1mm, The remaining 17 points have an error over 1mm. The result is that 65% of the time the system performed with an accuracy less than or equal to 0.5mm along the X-axis. Likewise, Figure 7.2 is a frequency plot of error along the Y-axis. It shows that 82% of the time the system performed with an accuracy of 0.5mm or better along the Y-axis. Table 7.1 displays the maximum, average, median, mode, variance, and sample standard deviation values of the error.

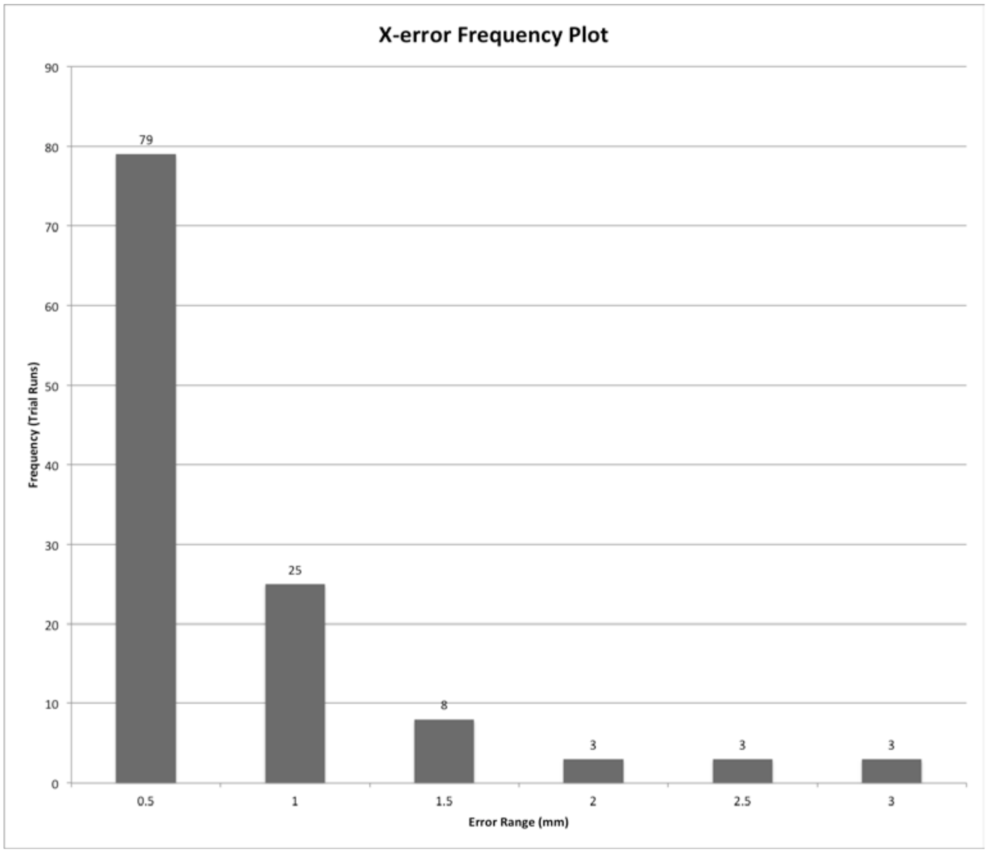


Figure 7.1 X-axis Error Frequency Plot

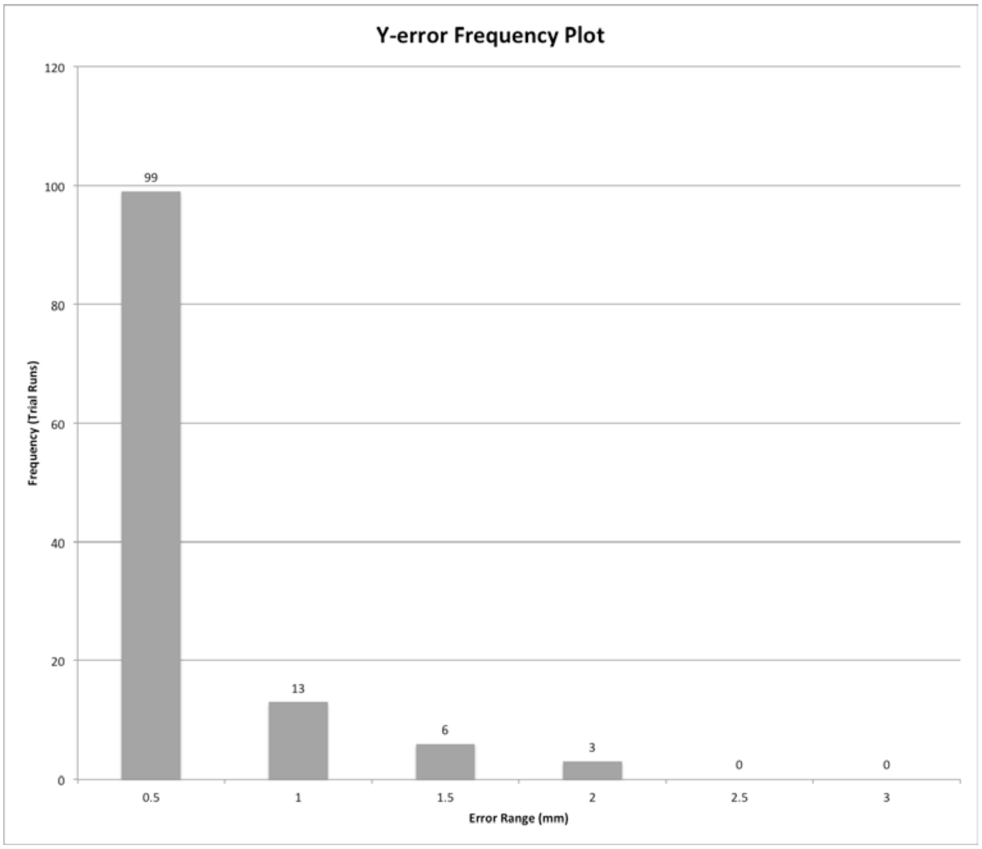
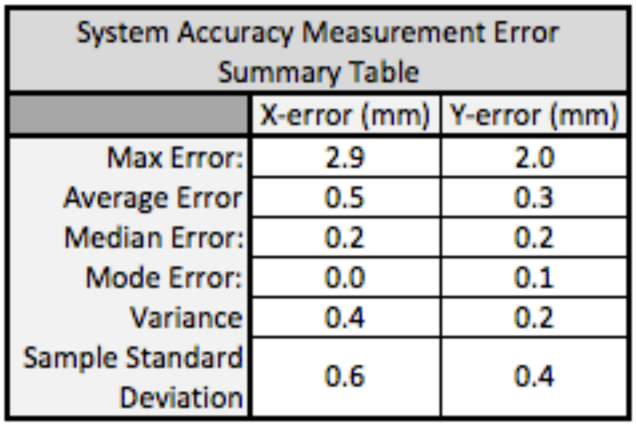


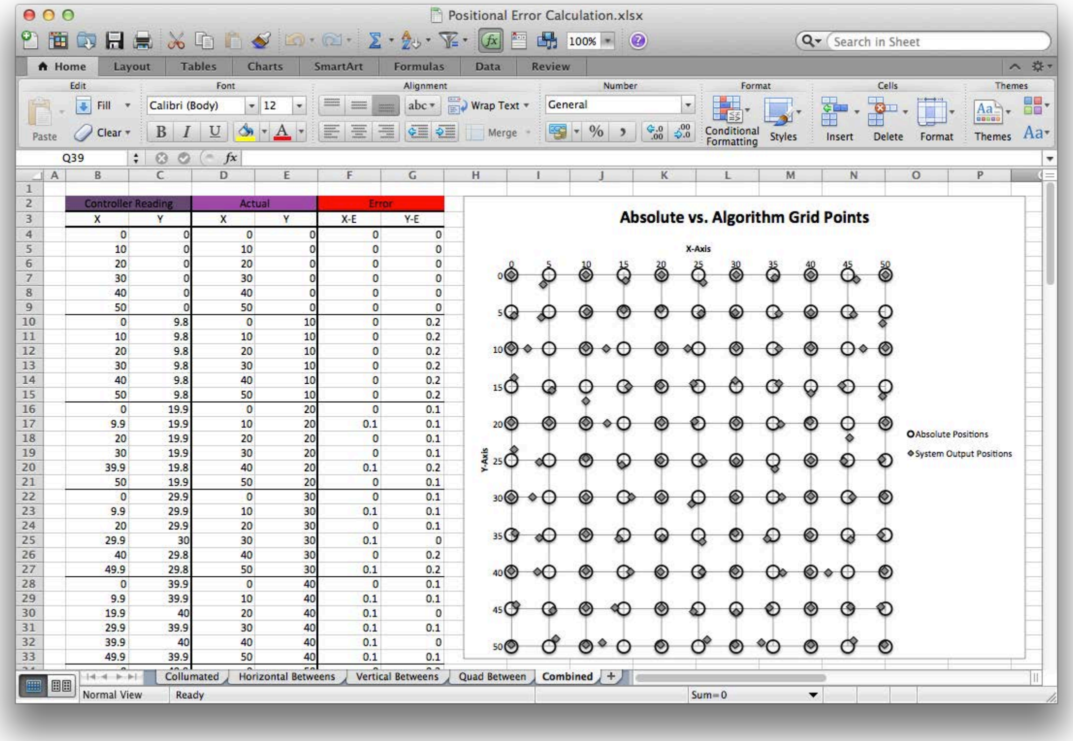
Figure 7.2 Y-axis Error Frequency Plot

Table 7.1 Maximum, Average, Median, Mode, Variance, and Sample Standard Deviation of Accuracy Measurement Error



System Accuracy Measurement Error Summary Table		
	X-error (mm)	Y-error (mm)
Max Error:	2.9	2.0
Average Error	0.5	0.3
Median Error:	0.2	0.2
Mode Error:	0.0	0.1
Variance	0.4	0.2
Sample Standard Deviation	0.6	0.4

Both the algorithm detected positions and the visually measured positions are recorded in a spreadsheet as shown in Figure 7.3 below to aid in plotting the position values and to calculate the error between the algorithm measurements and visual measurements.



Controller Reading		Actual		Error	
X	Y	X	Y	X-E	Y-E
0	0	0	0	0	0
10	0	10	0	0	0
20	0	20	0	0	0
30	0	30	0	0	0
40	0	40	0	0	0
50	0	50	0	0	0
0	9.8	0	10	0	0.2
10	9.8	10	10	0	0.2
20	9.8	20	10	0	0.2
30	9.8	30	10	0	0.2
40	9.8	40	10	0	0.2
50	9.8	50	10	0	0.2
0	19.9	0	20	0	0.1
9.9	19.9	10	20	0.1	0.1
20	19.9	20	20	0	0.1
30	19.9	30	20	0	0.1
39.9	19.8	40	20	0.1	0.2
50	19.9	50	20	0	0.1
0	29.9	0	30	0	0.1
9.9	29.9	10	30	0.1	0.1
20	29.9	20	30	0	0.1
29.9	30	30	30	0.1	0
40	29.8	40	30	0	0.2
49.9	29.8	50	30	0.1	0.2
0	39.9	0	40	0	0.1
9.9	39.9	10	40	0.1	0.1
19.9	40	20	40	0.1	0
29.9	39.9	30	40	0.1	0.1
39.9	40	40	40	0.1	0
49.9	39.9	50	40	0.1	0.1

Figure 7.3 Recorded Data for Accuracy Analysis

Figure 7.4 shows a plot containing true grid positions as open circles and the algorithm output of the laser at those positions as diamonds. Figure 7.5 shows a plot of the absolute X-error versus the absolute Y-error of the measurements shown in the Figure 7.4.

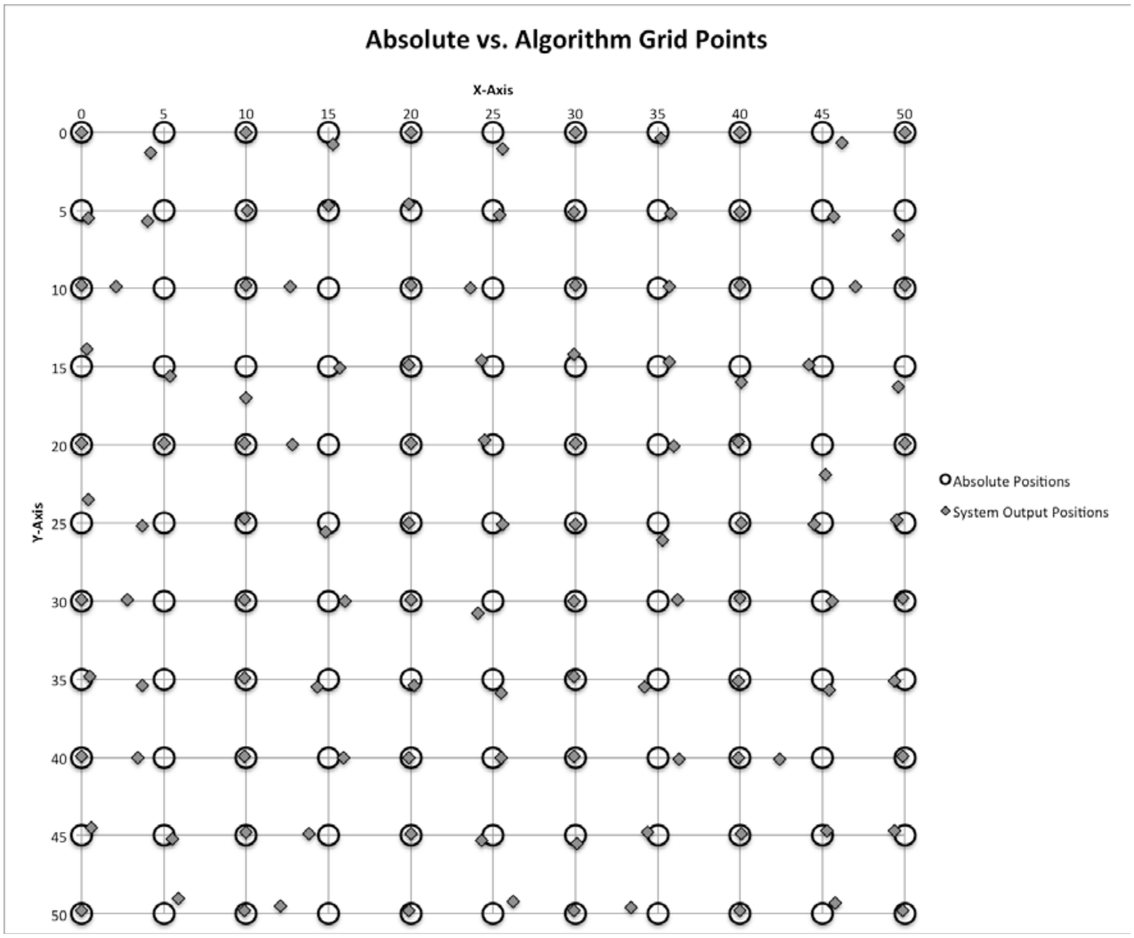


Figure 7.4 Grid Position Measurements

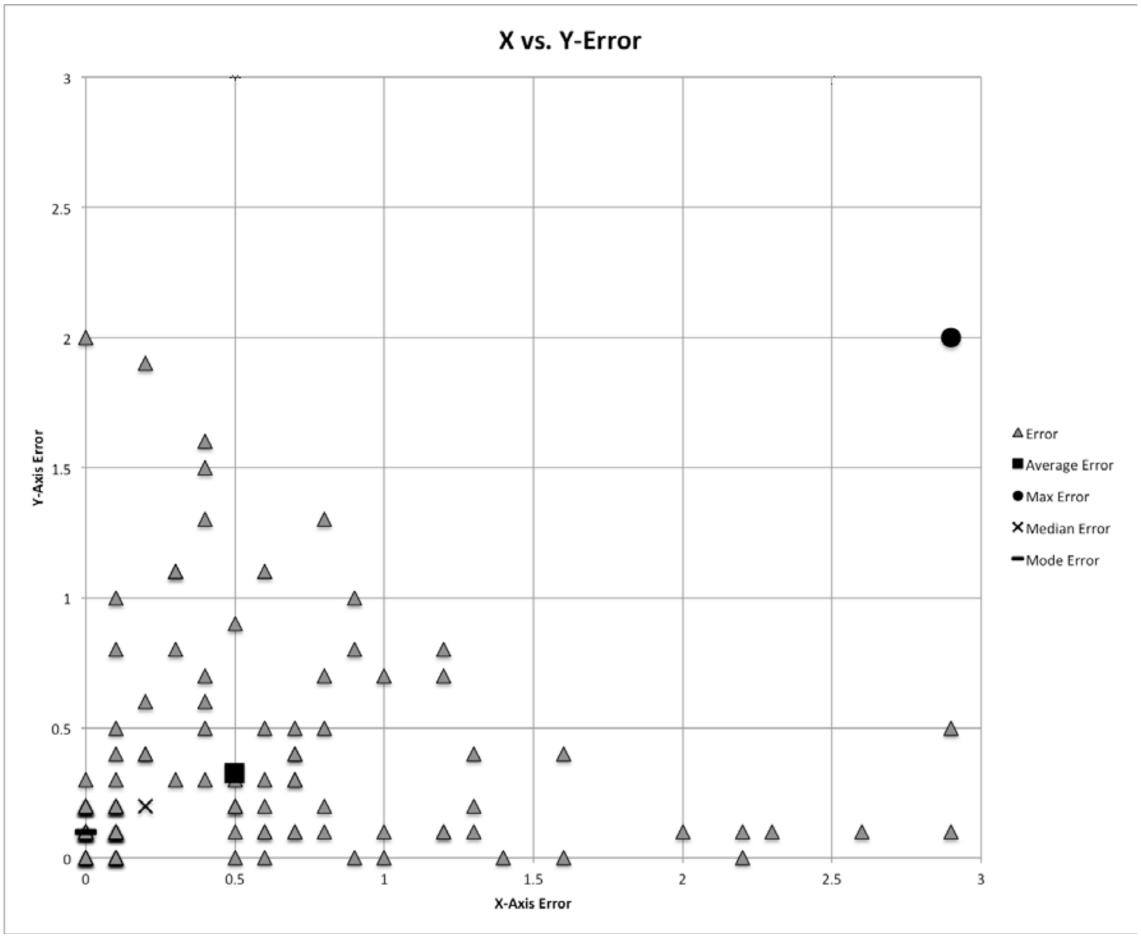


Figure 7.5 Grid Position Measurements Error Plot

From Figure 7.4, it can be seen that the system is the most accurate when directly above a photodiode. In that case, primarily only a single photodiode is detecting light, making the center-of-gravity calculation simple. When the laser is between photodiodes, the measurements are less accurate. The light is hitting multiple photodiodes, as well as the space between the photodiodes and the metallic leads of the photodiodes. The result is that some of the output light from the laser goes undetected between the photodiodes and some is scattered by the metal leads.

Despite the decreased accuracy between the photodiodes, Figure 7.5 shows that the system primarily functions in the ±1mm of error range with an even higher percentage in the ±0.5mm range. The triangle markers closest to zero show that there are multiple points stacked in this region. Therefore, the accuracy of the system is reasonable for a system with 10mm center-to-center grid spacing.

7.2 Test 2

7.2.1 Alignment Step 1 - Counterclockwise

The second test was to perform the full alignment process. Each step of the alignment was run with each laser position recorded by the algorithm, as well as measured via visual inspection. The results of both measuring methods are shown in the following pages. Measuring the two points to calculate the robot X-vector and then moving the lead laser to the Home Posiion is the first alignment step. Figure 7.6 shows the algorithm measurements of the lead laser for 30 runs with a small counterclockwise rotational misalignment between the laser plate and the photodiode plate. Figure 7.7 shows the visual measurements of the same 30 runs.

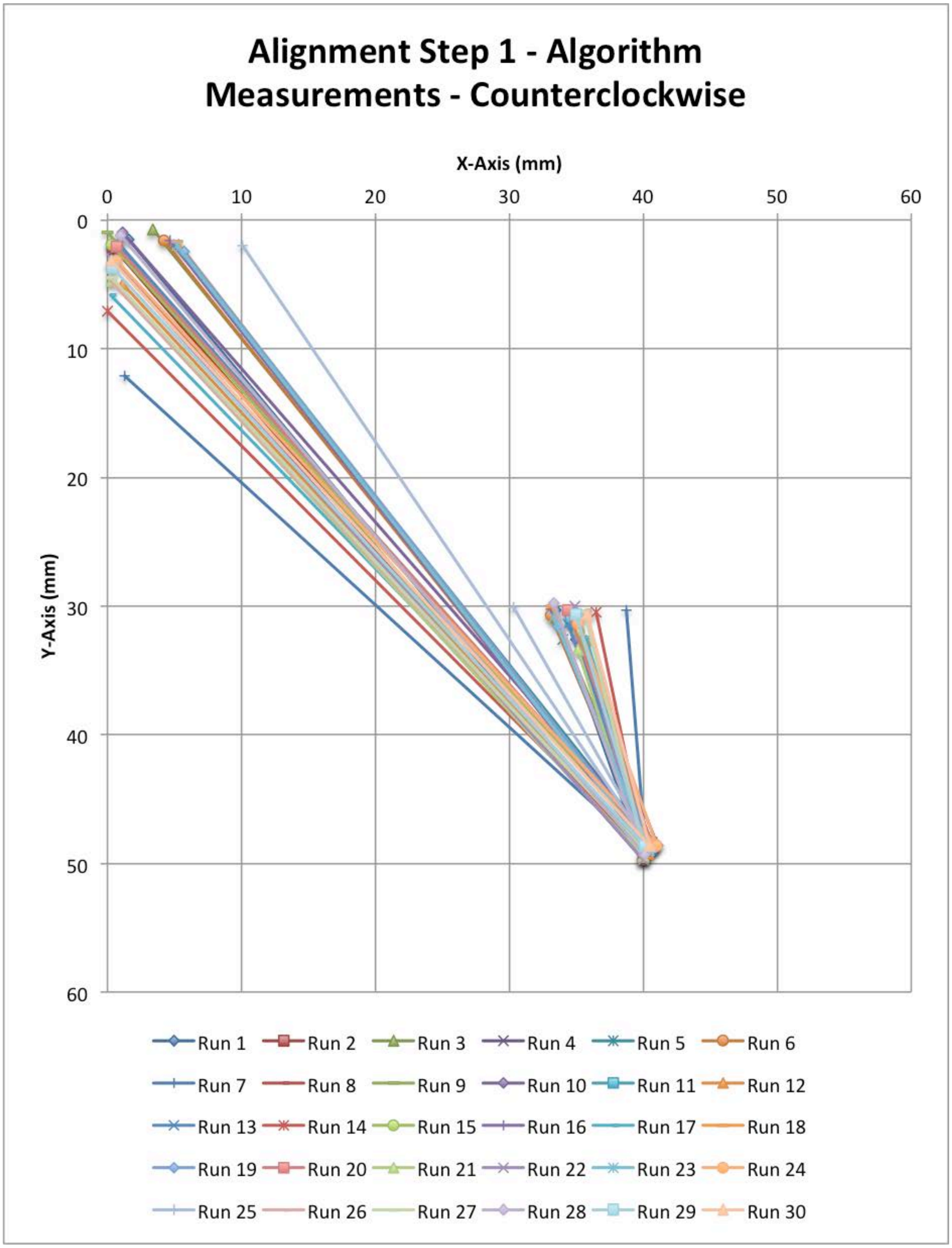


Figure 7.6 Step 1 Algorithm Alignment Measurements - Counterclockwise Rotation

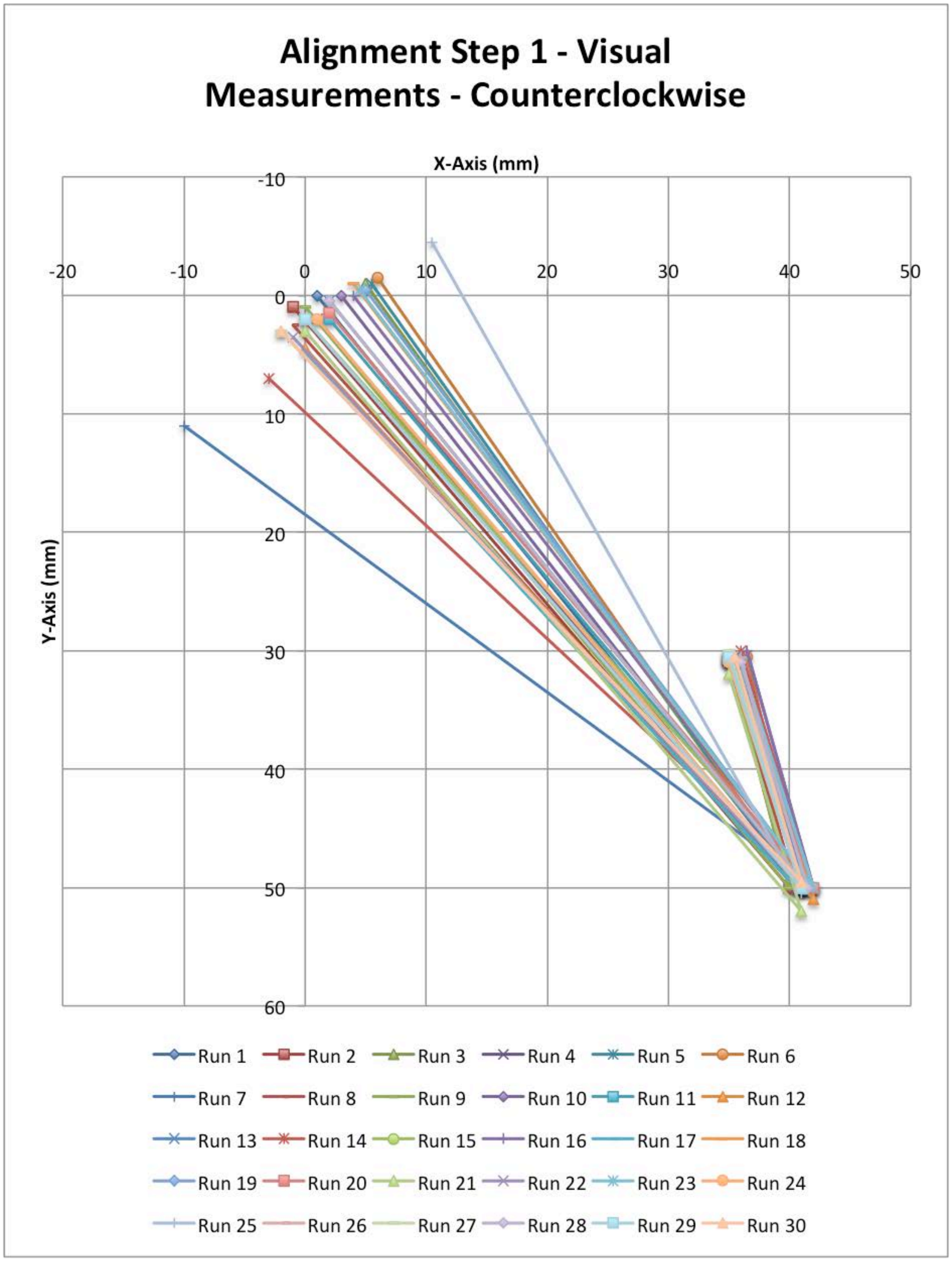


Figure 7.7 Step 1 Visual Alignment Measurements - Counterclockwise Rotation

The difference between Figures 7.6 and 7.7 show the difference between the calculated path and the actual path taken by the plate. Figure 7.6 shows what the computer position detection algorithm calculated, while Figure 7.7 shows what the actual position was (within ±0.5mm). The point that the lead laser ended in at the upper left of the plot is a result of a calculation made to move the laser to that position. The first two points on the right side of the plot show close correlation, because the observed positions were close together each time. The resulting points show less correlation because they are a result of the calculations made based on the points read in by the algorithm, therefore the error from the read-in values affects the result. When the algorithm reads in the laser locations, the error demonstrated above is introduced into the system. The values are then used to calculate the position of the robot X-vector in relation to the location of the grid X-vector. Therefore, error in the position detection of the two laser points causes the calculation to give an incorrect vector and create incorrect X and Y-movements, resulting in an incorrect position of the lead laser.

For the upper point on the right side of the Grid Array, the 30 runs have a maximum error of 11.3mm in the X-direction and 6.5mm in the Y-direction with an average error of 1.4mm in both the X and Y-directions. This error value is likely larger than the average error value found based on the first test described above because two of the other three lasers were close enough to the photodiode grid to cast small amounts of light onto photodiodes near the edge, reducing the accuracy of the X-vector measurement, and ultimately causing the system to incorrectly calculate the necessary movements.

Based on the X-vector calculated by moving to the two points shown on both Figure 7.6 and Figure 7.7, X and Y-movements in the robot reference frame are sent to the robot. These movements are calculated to get the lead laser to the home position (X = 0, Y = 0) of the Grid Array. The error in this process is high for one primary reason. Because of the error in the initial position detection of the two points used to calculate the X-vector of the robot, the calculations for the X and Y-movements of the robot are based on an incorrect vector. Once the commands are fed to the robot, the robot moves the distances along its actual X and Y-axes and ends up in an incorrect location.

Figure 7.6 shows the average end location of the lead laser closer to the home position than does Figure 7.7 because the Grid Array has no sense of a negative position. If the lead laser traveled beyond the Grid Array to the left or above the array, the Grid Array would use whatever scatter light was coming from the laser to calculate a position. Therefore, a laser off of the Grid Array just to the left of the plate would give an X-value of zero or close to zero because only the photodiodes near the edge would be detecting the light.

A visual representation of the error described above is explained using Figure 7.8. Both the algorithm and visual calculations for the alignment show error between the movements needed to align the lead laser to the home position, but the actual end location of the laser is different. As described above, the vector used to create the movement calculations is incorrect. Therefore, the path labeled Hybrid in the figure shows a better alignment path that is closer to the visual end position of the lead laser. The significance of this is that the calculation method is correct but produces incorrect results because it starts with incorrect input values. To find this point, the position adjustment calculations made by the system are applied to the visual vector, which is what the robot essentially travels along, therefore coming up with a plotted output that is closer to what happened in reality.

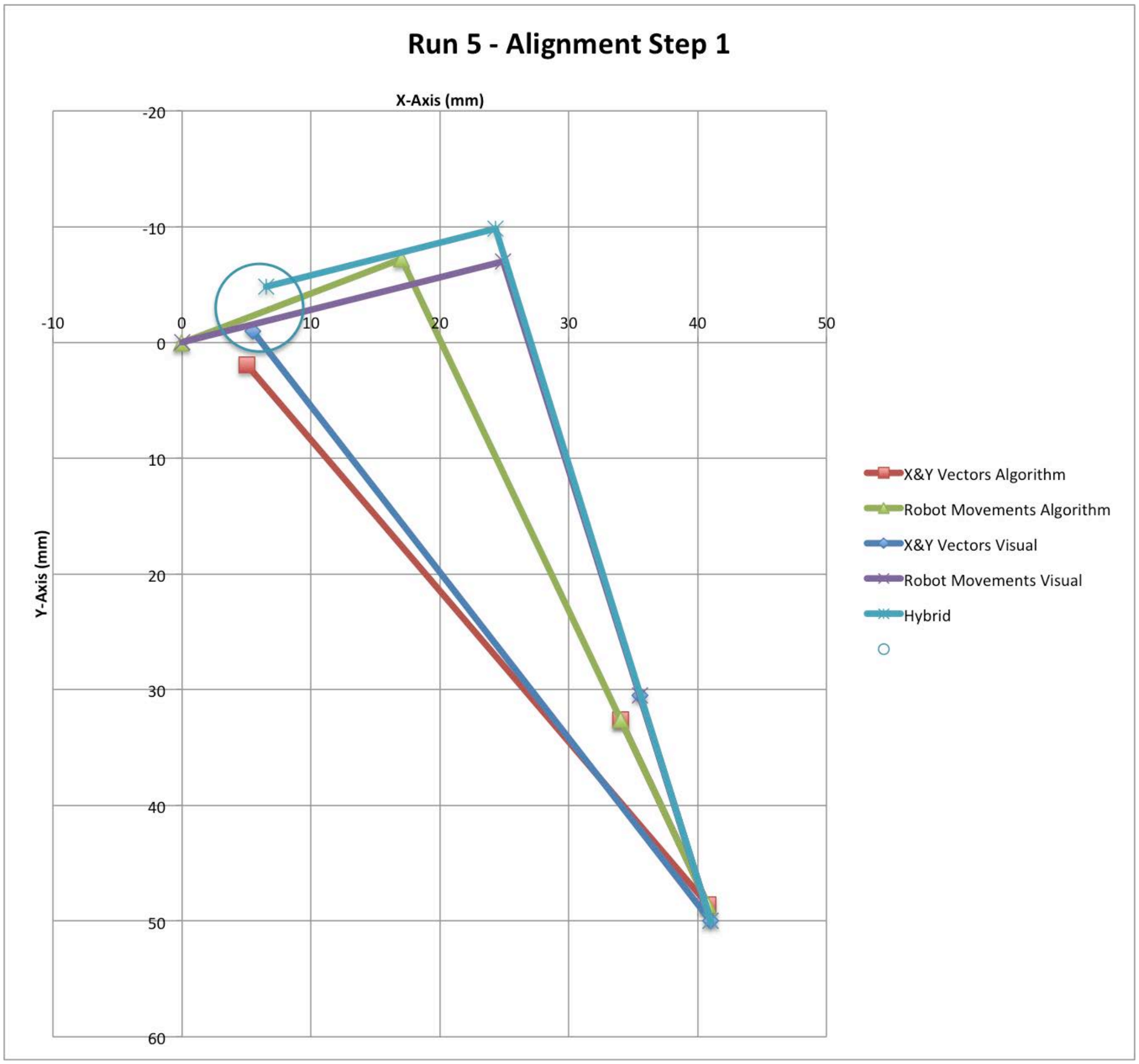
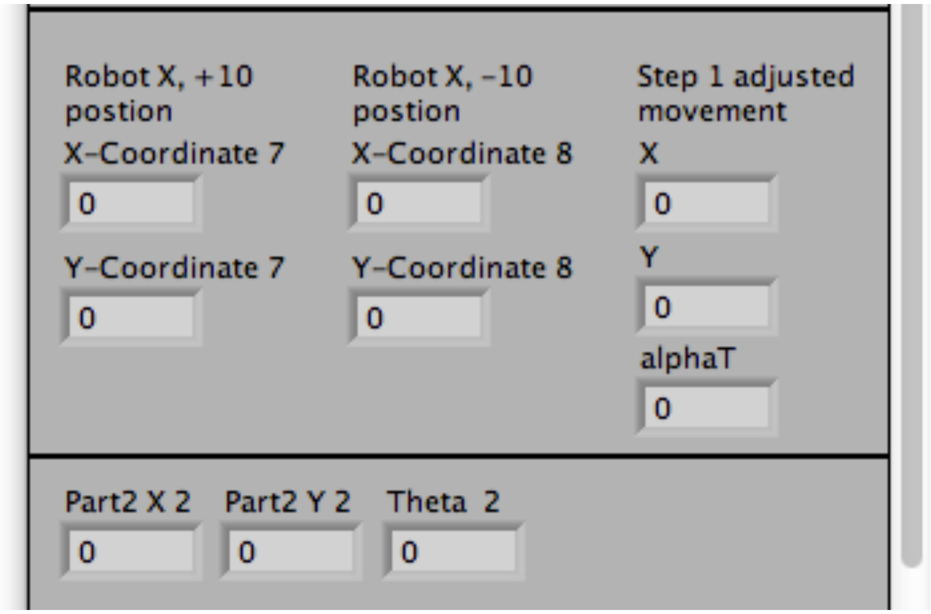


Figure 7.8 Run 5 Step 1 Alignment with Hybrid Path Calculation

The error in the positioning of the lead laser in the first step of the alignment is compensated in part by the second step of the alignment process. This step includes a feedback loop that continues to adjust the position of the robot with the four lasers until the center of the laser plate is within ±1.5mm of the center of the photodiode Grid Array. Therefore, if the initial alignment after the first adjustment is not within the error bounds, the position is updated until it is within the error bounds.

The computer measurements were output to the screen as shown in Figures 7.9 and 7.10. Figure 7.9 shows values in four different groups. The first group on the far left of the upper segment shows the X and Y-values once the robot has made the first movement. The middle group shows the X and Y-values of the second point once the robot has made the second movement. These two points are then used to calculate the X and Y-movements made in the robot reference frame to move the lead laser to the home position. The last group in the upper segment displays the calculated X and Y-movements as well as the angle between the robot X-vector and the Grid Array X-vector. Finally, the group in the lower segment represents the X and Y-movements and rotation performed by the robot to align the laser plate to the Grid Array after Step 1. These values are also recorded at the end of the alignment as the error value between the laser center and the robot center.



Robot X, +10 postion
X-Coordinate 7
0
Y-Coordinate 7
0

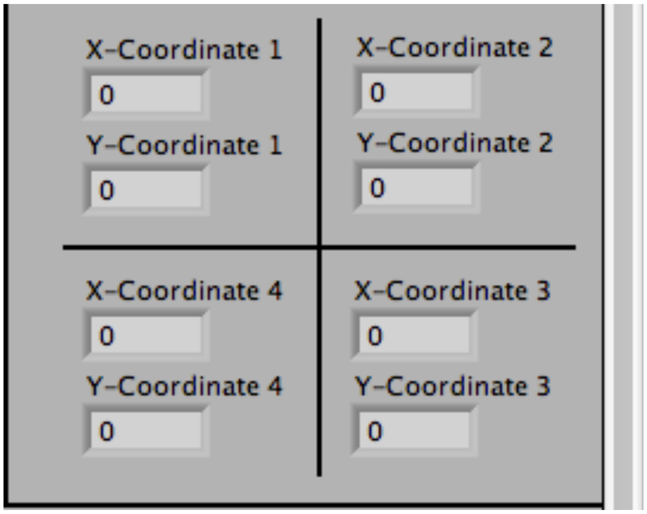
Robot X, -10 postion
X-Coordinate 8
0
Y-Coordinate 8
0

Step 1 adjusted movement
X
0
Y
0
alphaT
0

Part2 X 2
0
Part2 Y 2
0
Theta 2
0

Figure 7.9 Laser Position in Each of the Two Points Used in Calculating X-Vector and Translational and Rotational Values sent to Robot for Alignment

In Step 2 the position of each of the four lasers is detected before each adjustment. The coordinate values are displayed in the fields shown in Figure 7.10 below. To analyze the performance of the system, values from each step and each run of the system are recorded to measure accuracy and error of the system, as well as to generate the plots shown in this section. Values are recorded in an excel sheet, like that shown in Figure 7.12, and then used to generate the plots shown in this chapter.



X-Coordinate 1: 0
Y-Coordinate 1: 0
X-Coordinate 2: 0
Y-Coordinate 2: 0
X-Coordinate 4: 0
Y-Coordinate 4: 0
X-Coordinate 3: 0
Y-Coordinate 3: 0

Figure 7.10 Laser Coordinates in Each Quadrant Relative to Home Position

The visually recorded data was collected via inspection of the laser positions on the grid. Figure 7.11 shows an example photo of the lasers on the grid and their relative orientation using a sketch overlaid on the image.

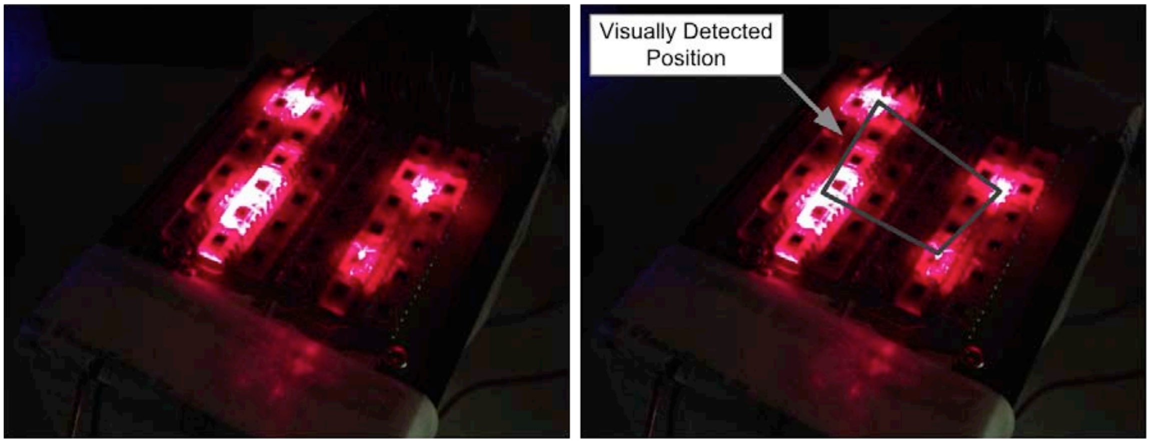
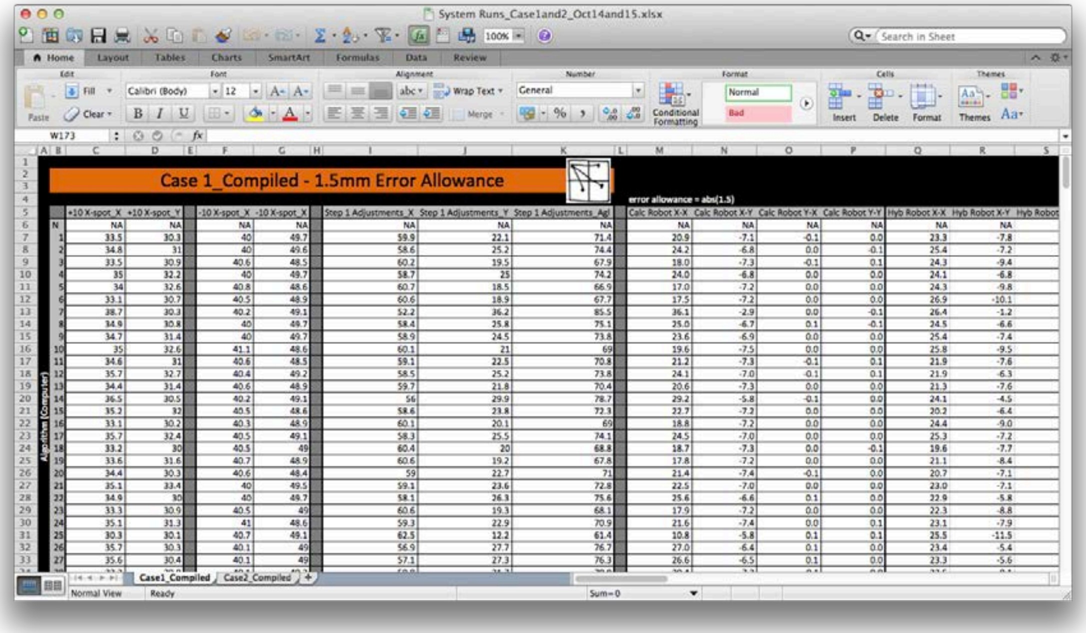


Figure 7.11 Photo of Lasers on the Grid Array and an Identical Photo with a Sketch Overlaid on the Image



Case 1_Compiled - 1.5mm Error Allowance

error allowance = abs(1.5)

N	+10 X-spot_X	+10 X-spot_Y	-10 X-spot_X	-10 X-spot_X	Step 1 Adjustments_X	Step 1 Adjustments_Y	Step 1 Adjustments_Agl	Calc Robot X-X	Calc Robot X-Y	Calc Robot Y-X	Calc Robot Y-Y	Hyb Robot X-X	Hyb Robot X-Y	Hyb Robot
	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	
1	33.5	30.3	40	49.7	59.9	22.1	71.4	20.9	-7.1	-0.1	0.0	23.3	-7.8	
2	34.8	31	40	49.6	58.6	25.2	74.4	24.2	-6.8	0.0	-0.1	25.4	-7.2	
3	33.5	30.9	40.6	48.5	60.2	19.5	67.9	18.0	-7.3	-0.1	0.1	24.3	-9.4	
4	35	32.2	40	49.7	58.7	25	74.2	24.0	-6.8	0.0	0.0	24.1	-6.8	
5	34	32.6	40.8	48.6	60.7	18.5	66.9	17.0	-7.2	0.0	0.0	24.3	-9.8	
6	33.1	30.7	40.5	48.9	60.6	18.9	67.7	17.5	-7.2	0.0	0.0	26.9	-10.1	
7	38.7	30.3	40.2	49.1	52.2	36.2	85.5	36.1	-2.9	0.0	-0.1	26.4	-1.2	
8	34.9	30.8	40	49.7	58.4	25.8	75.1	25.0	-6.7	0.1	-0.1	24.5	-6.6	
9	34.7	31.4	40	49.7	58.9	24.5	73.8	23.6	-6.9	0.0	0.0	25.4	-7.4	
10	35	32.6	41.1	48.6	60.1	21	69	19.6	-7.5	0.0	0.0	25.8	-9.5	
11	34.6	31	40.6	48.5	59.1	22.5	70.8	21.2	-7.3	-0.1	0.1	21.9	-7.6	
12	35.7	32.7	40.4	49.2	58.5	25.2	73.8	24.1	-7.0	-0.1	0.1	21.9	-6.3	
13	34.4	31.4	40.6	48.9	59.7	21.8	70.4	20.6	-7.3	0.0	0.0	21.3	-7.6	
14	36.5	30.5	40.2	49.1	56	29.9	78.7	29.2	-5.8	-0.1	0.0	24.1	-4.5	
15	35.2	32	40.5	48.6	58.6	23.8	72.3	22.7	-7.2	0.0	0.0	20.2	-6.4	
16	33.1	30.2	40.3	48.9	60.1	20.1	69	18.8	-7.2	0.0	0.0	24.4	-9.0	
17	35.7	32.4	40.5	49.1	58.3	25.5	74.1	24.5	-7.0	0.0	0.0	25.3	-7.2	
18	33.2	30	40.5	49	60.4	20	68.8	18.7	-7.3	0.0	-0.1	19.6	-7.7	
19	33.6	31.6	40.7	48.9	60.6	19.2	67.8	17.8	-7.2	0.0	0.0	21.1	-8.4	
20	34.4	30.3	40.6	48.4	59	22.7	71	21.4	-7.4	-0.1	0.0	20.7	-7.1	
21	35.1	33.4	40	49.5	59.1	23.6	72.8	22.5	-7.0	0.0	0.0	23.0	-7.1	
22	34.9	30	40	49.7	58.1	26.3	75.6	25.6	-6.6	0.1	0.0	22.9	-5.8	
23	33.3	30.9	40.5	49	60.6	19.3	68.1	17.9	-7.2	0.0	0.0	22.3	-8.8	
24	35.1	31.3	41	48.6	59.3	22.9	70.9	21.6	-7.4	0.0	0.1	23.1	-7.9	
25	30.3	30.1	40.7	49.1	62.5	12.2	61.4	10.8	-5.8	0.1	0.1	25.5	-11.5	
26	35.7	30.3	40.1	49	56.9	27.7	76.7	27.0	-6.4	0.1	0.0	23.4	-5.4	
27	35.6	30.4	40.1	49	57.1	27.3	76.3	26.6	-6.5	0.1	0.0	23.3	-5.6	

Figure 7.12 Data Input and Plotting Generation Sheet

7.2.2 Alignment Step 2 - Counterclockwise

After alignment of Part 1 is complete, the lead laser is at or near the Home Position, and the remaining three lasers are in an unknown orientation on the grid. Two cases are used to test the alignment capabilities of the system using all four lasers. One begins with the laser module at a small counterclockwise rotational misalignment relative to the grid and the other begins with the laser module at a small clockwise rotational misalignment. The rotations are kept small to ensure that all four lasers will be over the Grid Array once the lead laser is aligned with the Home Position.

Figure 7.13 below shows the algorithm-detected positions of the four lasers after the 30 Step 1 alignment runs and before Step 2 is performed. Because the algorithm cannot detect a negative number, all lead laser X and Y-position values are shown as positive or zero. In reality, the lead laser did travel past the home position on many occasions. The observed position of the lasers for each of the 30 runs is shown in Figure

7.14. In this figure, a more accurate location of the lead laser after the Step 1 alignment can be seen.

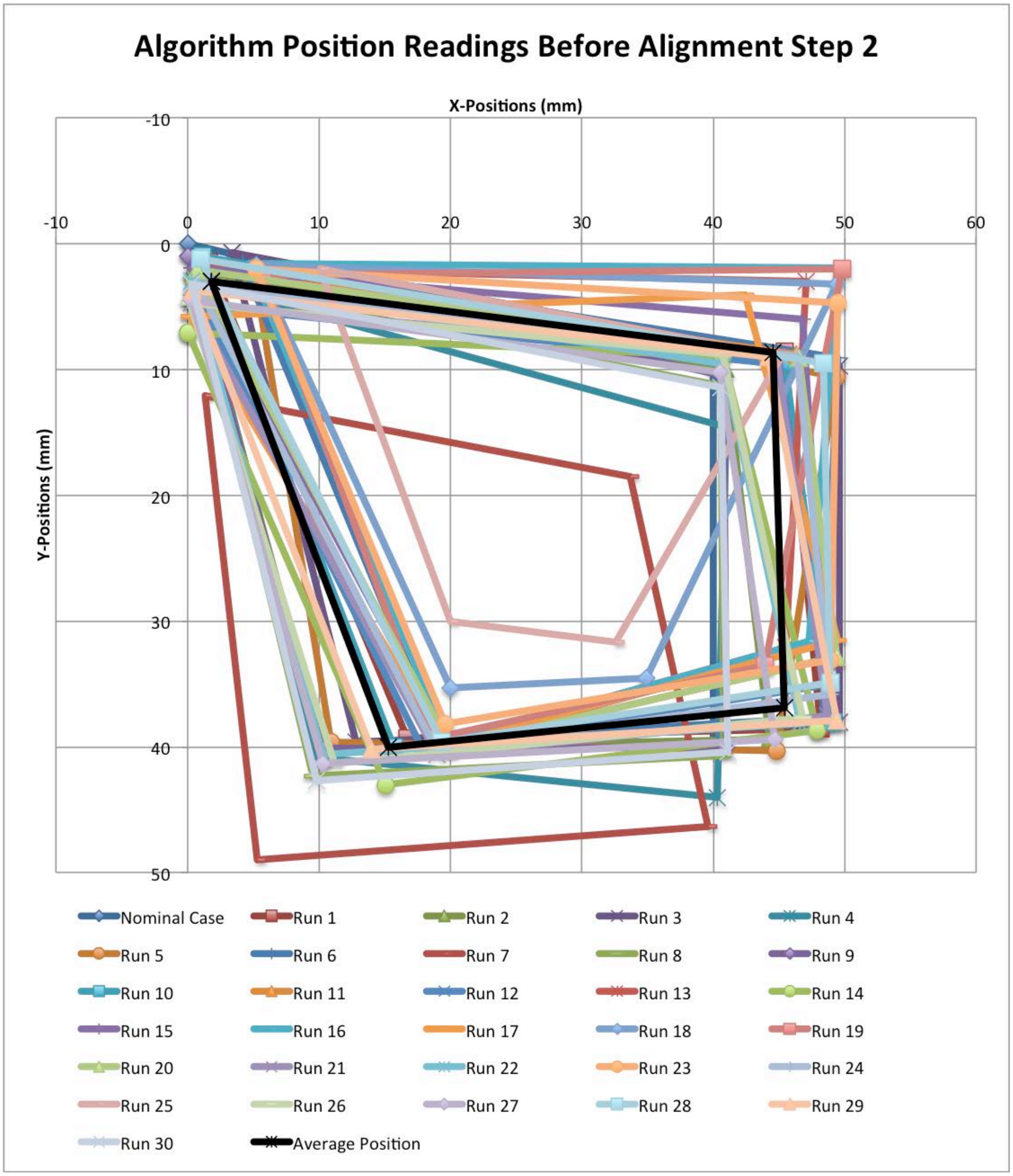


Figure 7.13 Algorithm Reading Laser Positions After Step 1 - 30 Runs

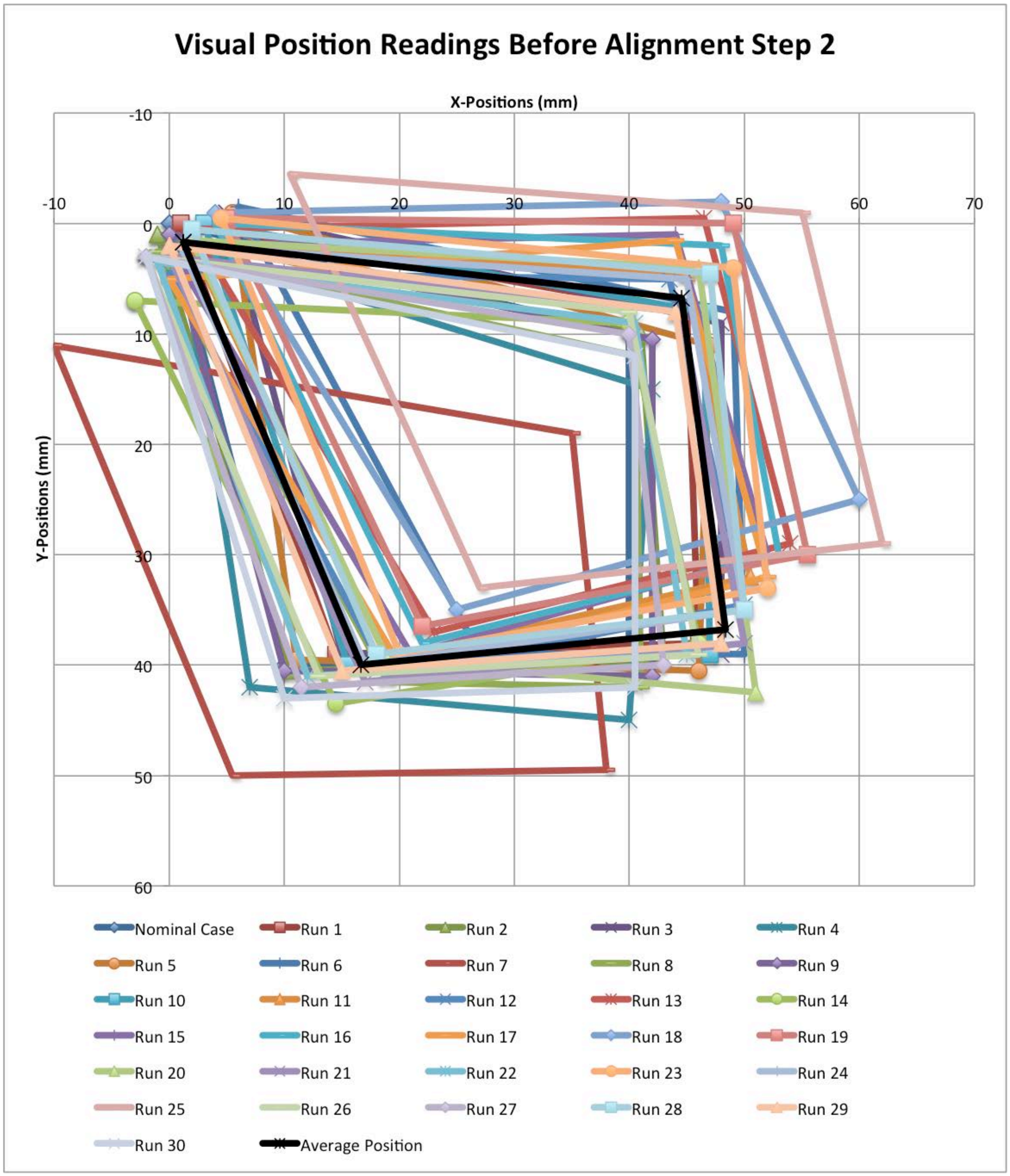


Figure 7.14 Visual Reading Laser Positions After Step 1 - 30 Runs

In the case shown above in which the plate started with a small counterclockwise misalignment, there are three instances where the lead laser is especially far from the home position. Because of this initial misalignment, it is difficult for the system to align the laser plate. It moved the laser plate until it read that it was within the error tolerance, but the visual results show that this was not the case. The reason for this is explained below.

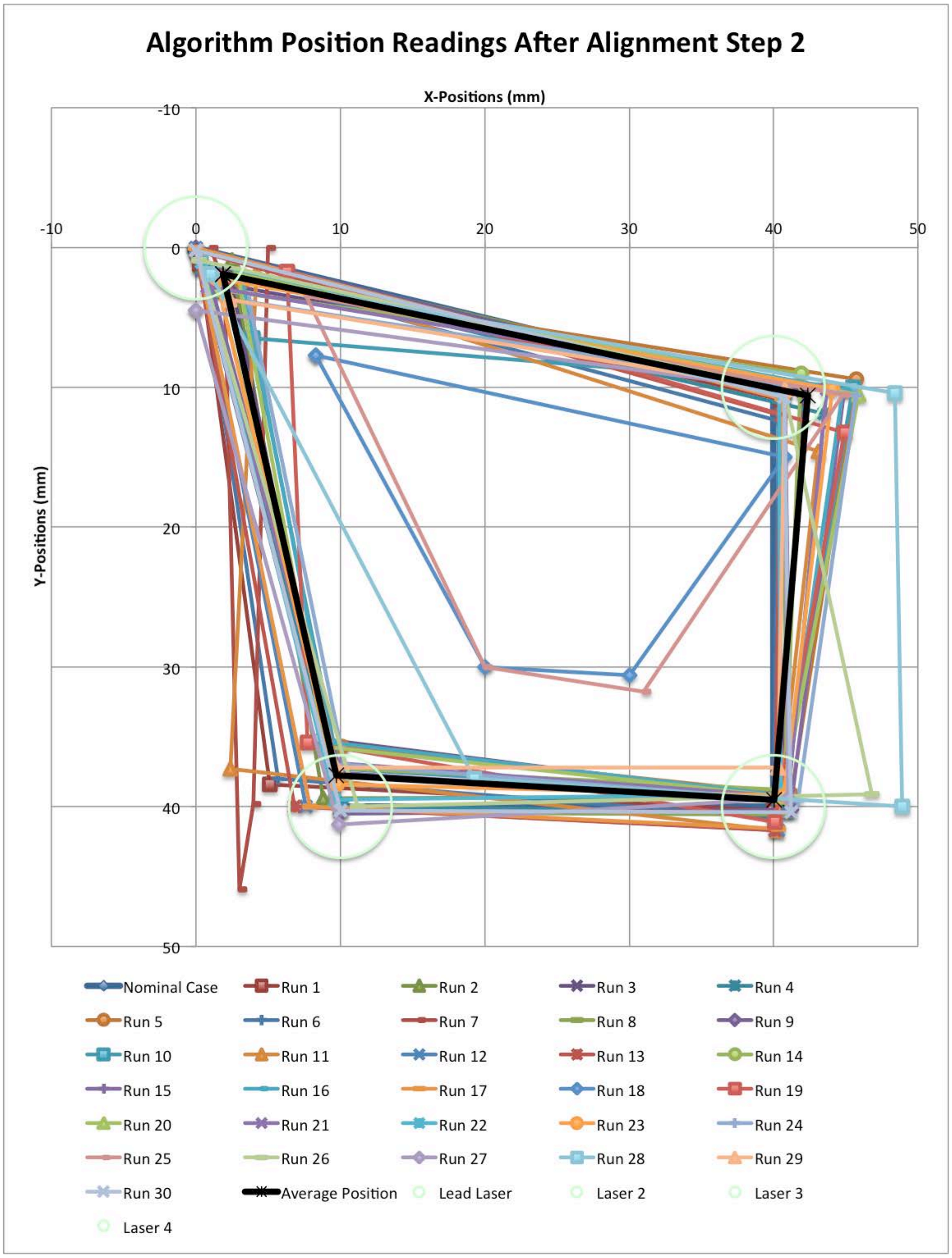


Figure 7.15 Algorithm Reading Laser Positions After Step 2 - 30 Runs

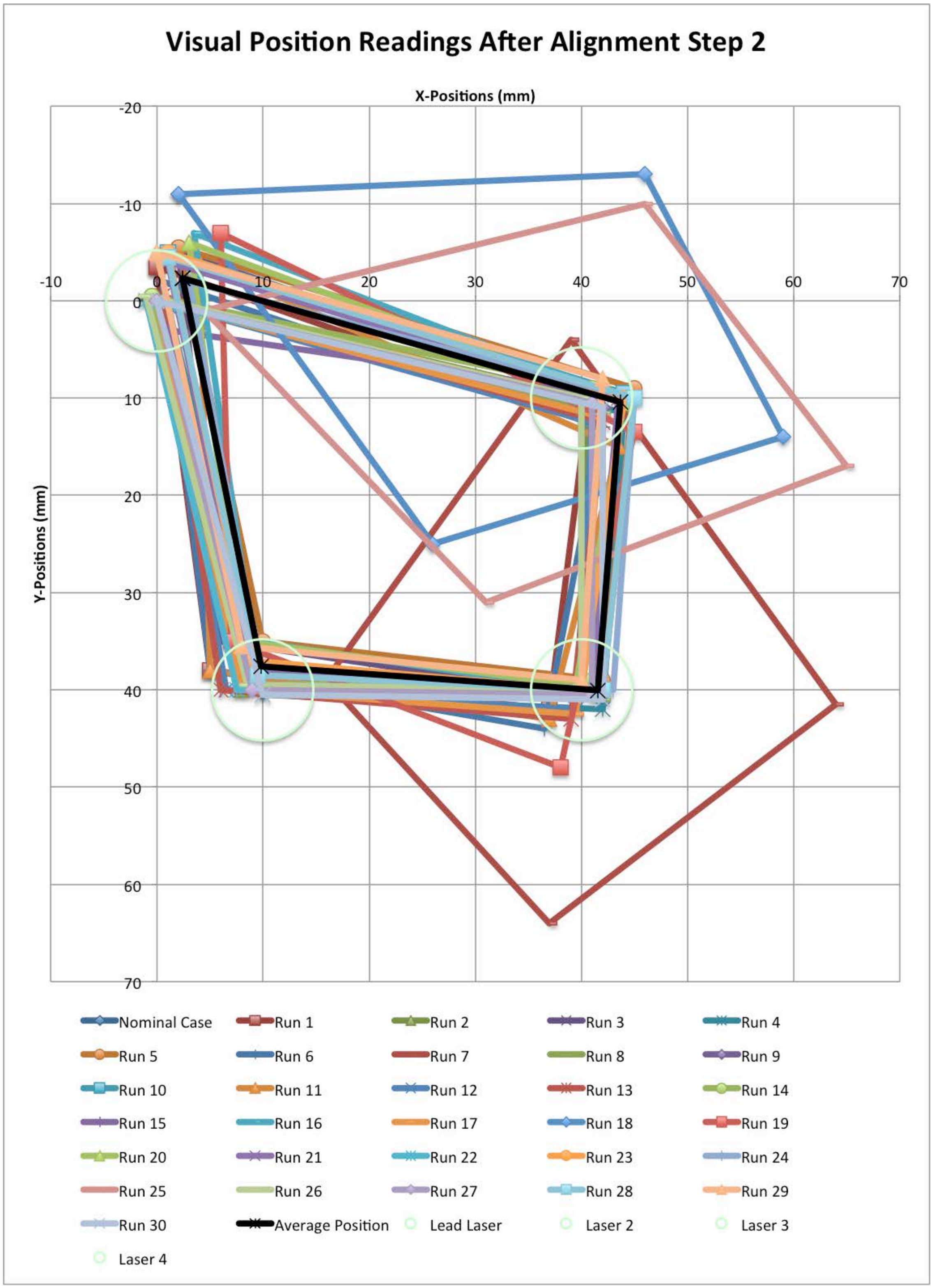


Figure 7.16 Visual Reading Laser Positions After Step 2 - 30 Runs

Figures 7.15 and 7.16 show the resulting location of the lasers after the alignment feedback loop of Step 2 has run. Figure 7.15 shows the algorithm readings of the location of each of the four lasers for each of the 30 runs. Similarly, Figure 7.16 shows the visual readings of the location of each of the four lasers for each of the 30 runs. It can been seen in Figure 7.16 that the same runs shown in Figure 7.14 that had large misalignments at the end of Step 1, also have large misalignment at the end of Step 2 as well. The visual version shows that the error was beyond the bounding limit. This is not as obvious in Figure 7.15 because of the way that the grid detects the position of each laser.

To detect four lasers at once, the grid is broken into quadrants containing nine photodiodes per quadrant. Each quadrant has a three-by-three grid that it uses to detect a laser. It performs a center-of-gravity type calculation like that discussed in Chapter 6. In the ideal scenario there is light from only one laser diode hitting any of the nine photodiodes in a quadrant. What happens in practice, especially when there is a large rotational or positional misalignment like that seen in three of the runs shown in Figure 7.14, is that light from another laser diode falls into the neighboring quadrant, altering both the detected position of that laser diode in its own quadrant and the detected position of the laser diode in the other neighboring quadrant. For example, if a laser diode is beyond the bounds of the Grid Array, but close enough to still shed some light onto the edge photodiodes, it is possible that another laser diode will be in a position that is close to the boundary between quadrants and also shed light into the quadrant of the aforementioned laser diode. Because of the small amount of light being detected from either edge of the quadrant, the center-of-gravity calculation will output a value that is close to the center of the quadrant because it is averaging out the light that it receives from the entire quadrant. Then, because some of the light from the other laser diode is not getting into its own quadrant, this quadrant will show a reading of the laser that is closer to the center than it actually is because the light passing into the neighboring quadrant is not calculated into the center-of-gravity calculation. This is portrayed in Figure 7.17 below.

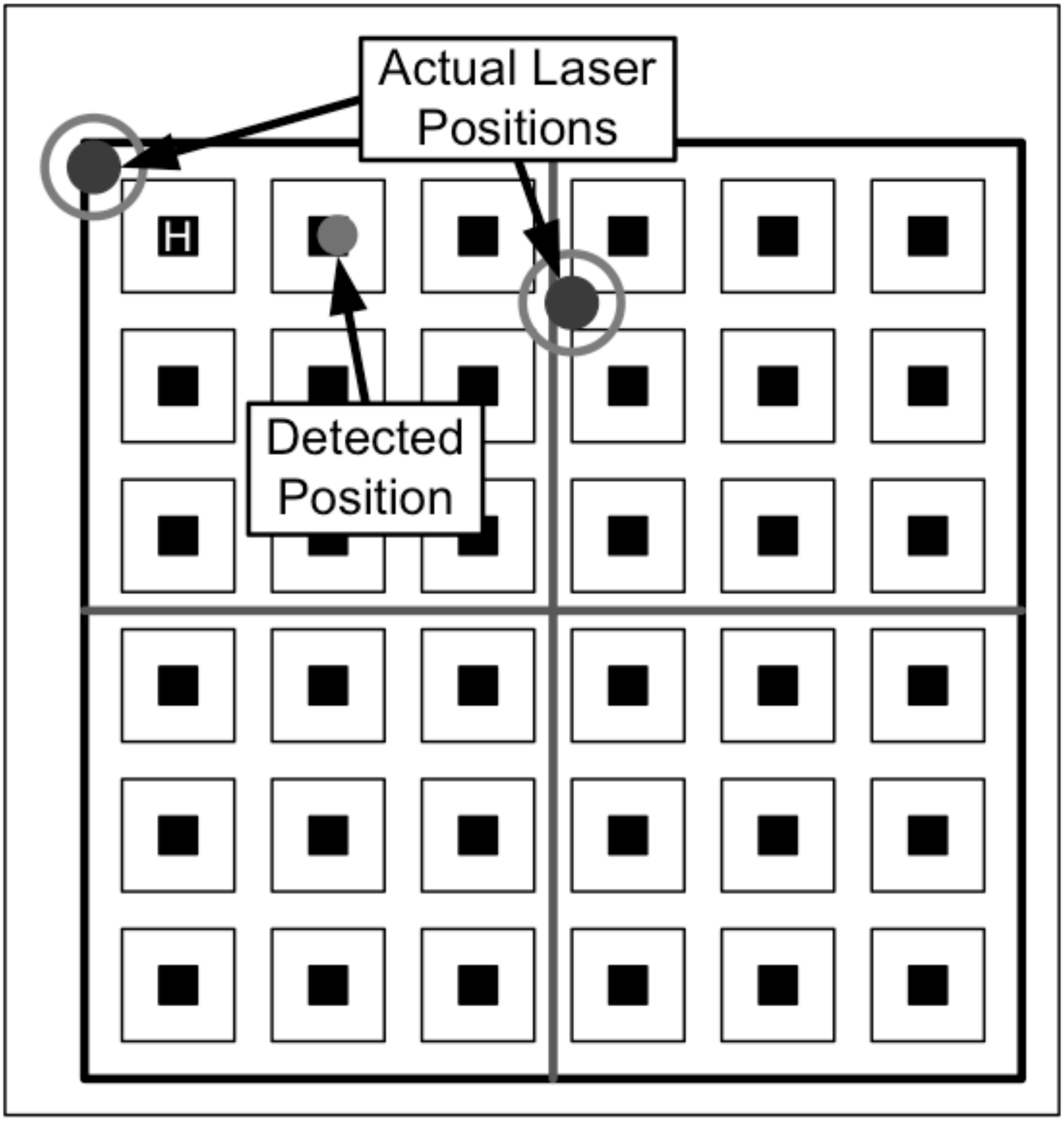


Figure 7.17 Center-of-Gravity Calculation Shortcoming Demonstration

As can be seen in Figure 7.16, the visual measurements show that in general the alignment was close to the desired position. The center of each circle on the corners represent where each laser should fall in an ideal and fully-aligned scenario. It can been seen in Figure 7.16 that the alignment was better than that shown in Figure 7.15.

In some scenarios, like those shown by the three misaligned runs in Figures 7.15 and 7.16 above, the system will detect that the error is within the ±1.5mm tolerance when in reality it is not. This is a result of the method of calculating the error between the ideal aligned state of the laser plate and the actual position of the plate. The distance between the detected position of each laser is calculated by finding the equation of the line between the lead laser and its corresponding diagonal laser, and by finding the equation of the line between the other two remaining lasers. The point of intersection of these two lines is calculated and used as center point of the lasers. The difference in X and Y-directions between this point and the center of the Grid Array is the error measure. However, because the position calculation of the four lasers is not always correct, as discussed above, the center position is not always correct either. For example, because light from one laser diode is striking an incorrect quadrant, the system may see the two laser diodes closer together than they really are. Figure 7.18 demonstrates the diagonals of the laser positions and the center point of both the Grid Array and the laser positions based on the algorithm reading after the second alignment step for the first trial run. Figure 7.19 represents the same thing but using the visual laser locations.

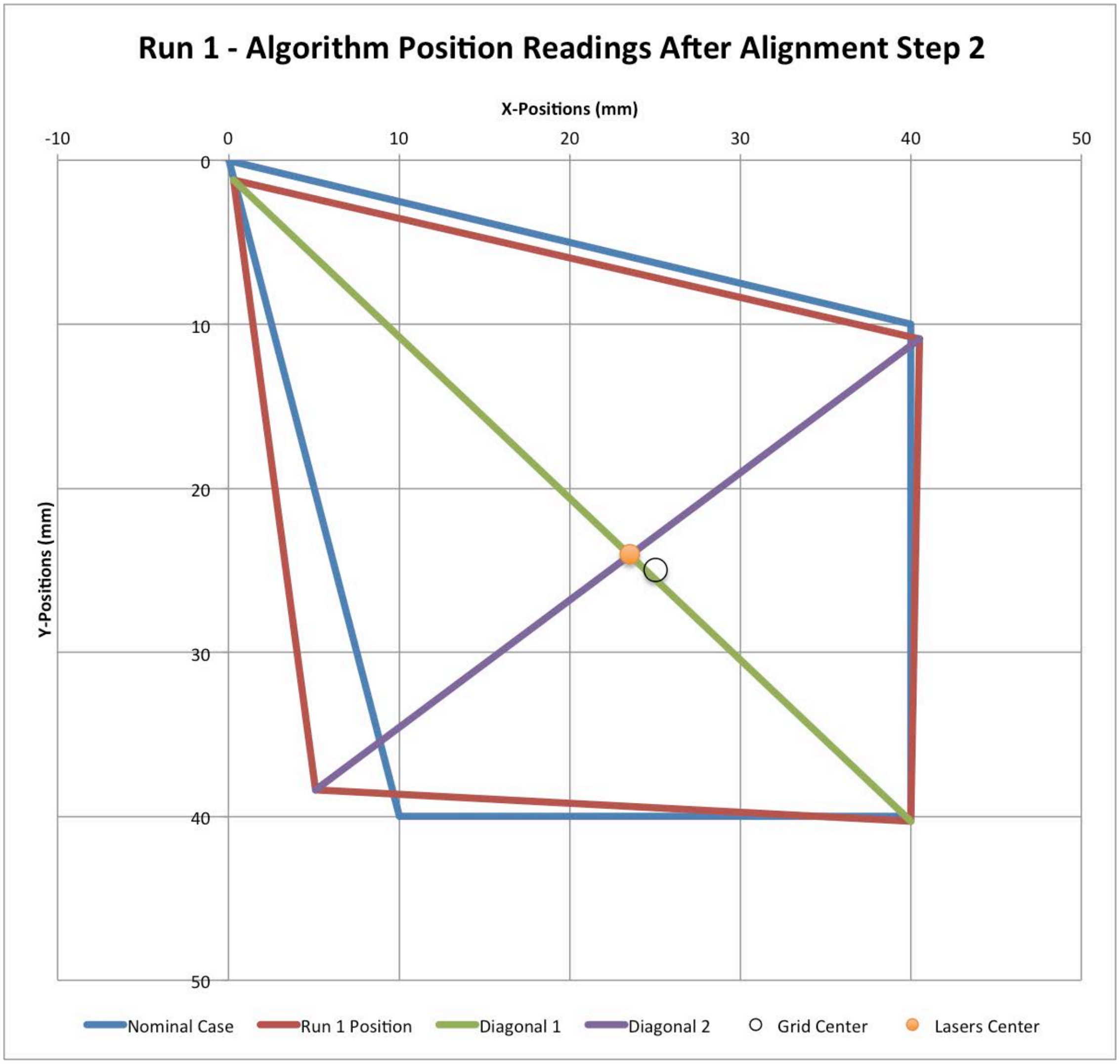


Figure 7.18 Run 1 Aligned Plate Center Position - Algorithm Reading

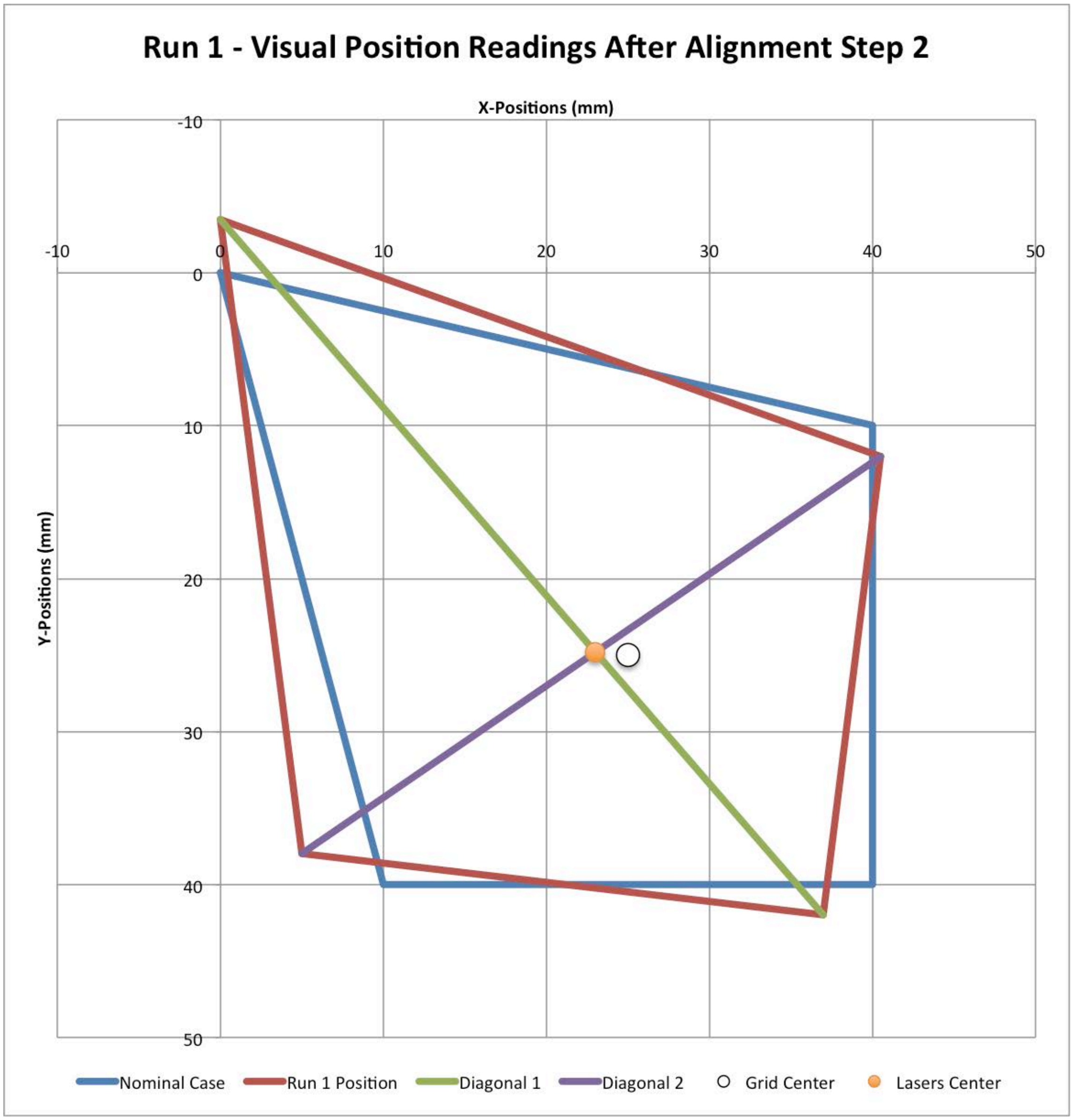


Figure 7.19 Run 1 Aligned Plate Center Position - Visual Reading

The dots in Figure 7.18 show a final absolute X-error for Run 1 of approximately 1.5mm and a final absolute Y-error of approximately 1mm, and the algorithm calculations show an X-error of 0.9mm and Y-error of 1.5mm. The two dots in Figure 7.19 show an X-error of approximately 2mm and a Y-error of approximately 0.2mm.

The maximum absolute error after Step 1 alignment is 11.3mm in the X-direction with an average error of 1.4mm and a maximum error of 6.5mm in the Y-direction with an average error of 1.4mm. After alignment Step 2 the maximum absolute error is 1.4mm in the X-direction with an average error of 0.5mm and a maximum error of 1.5mm in the Y-direction with an average error of 0.6mm. Plots of the alignment error after both Step 1 and Step 2 are shown in Figures 7.20 and 7.21 respectively.

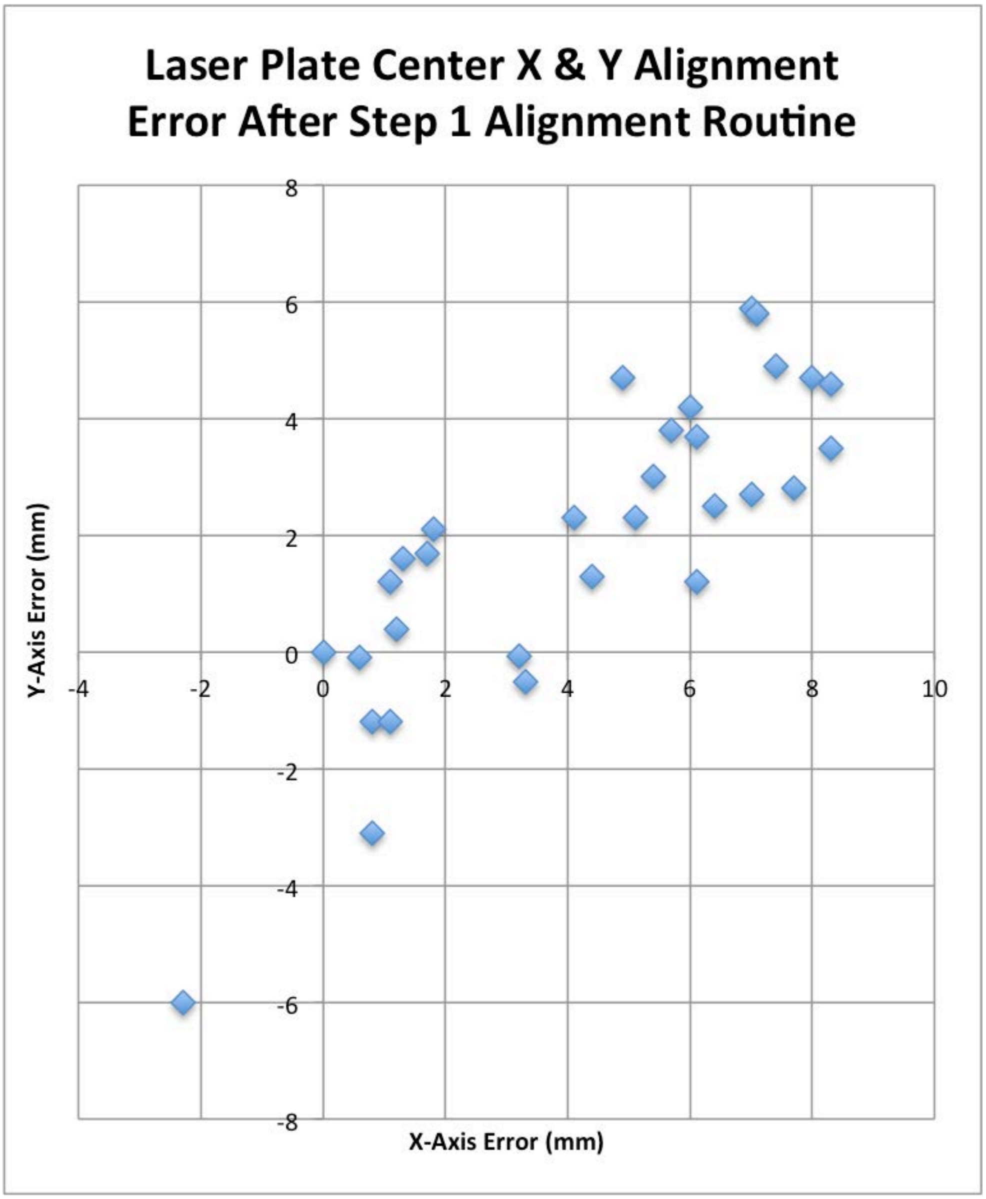


Figure 7.20 Center Point Alignment Error After Step 1

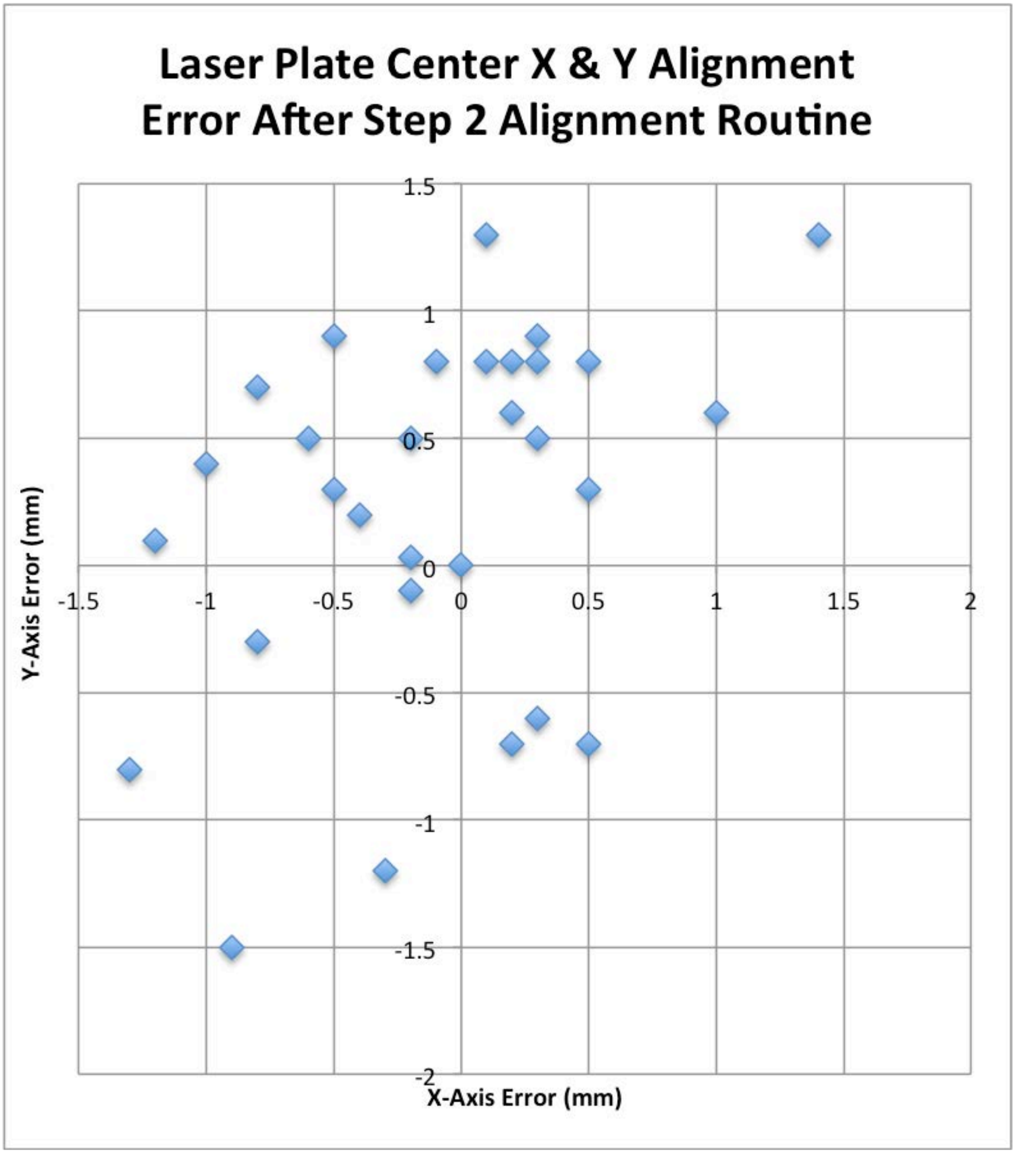


Figure 7.21 Center Point Alignment Error After Step 2

Due to the error introduced into the system in the alignment as mentioned above, the position of the lead laser is on average more spread out and further from the home position after Step 2 of the alignment. This is not surprising when considering the rotation and translation steps that the system undergoes to align the center of the laser plate to the center of the Grid Array. Figures 7.22 and 7.23 demonstrate this change, with Figure 7.22 showing the positions of the lead laser for each of the 30 runs as measured by the algorithm at the end of Step 1 next to the picture of the position of the lead laser at the end of Step 2. This same concept using visual inspection is shown in Figure 7.23.

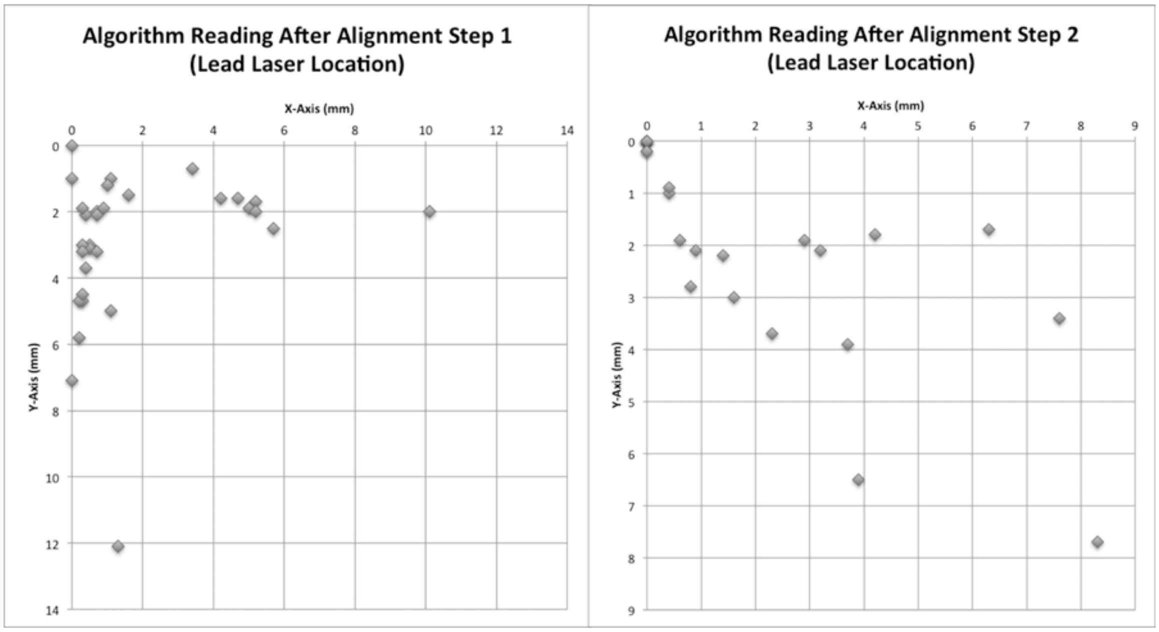


Figure 7.22 Lead Laser Position Before and After Step 2 Alignment - Algorithm Reading

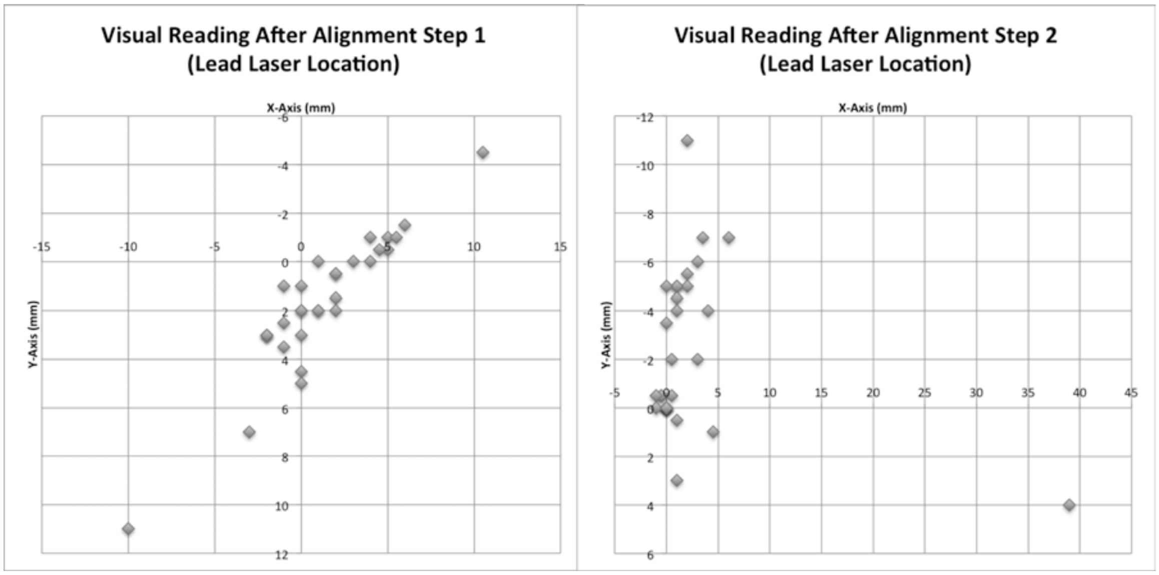


Figure 7.23 Lead Laser Position Before and After Step 2 Alignment - Visual Reading

7.2.3 Alignment Step 1 - Clockwise

The 30 runs starting with a small clockwise misalignment show similar results to those discussed above. Figures 7.24 - 7.34 and 7.37 - 7.38 show corresponding plots to those shown in Figures 7.6 - 7.8, 7.13 - 7.16, and 7.18 - 7-23.

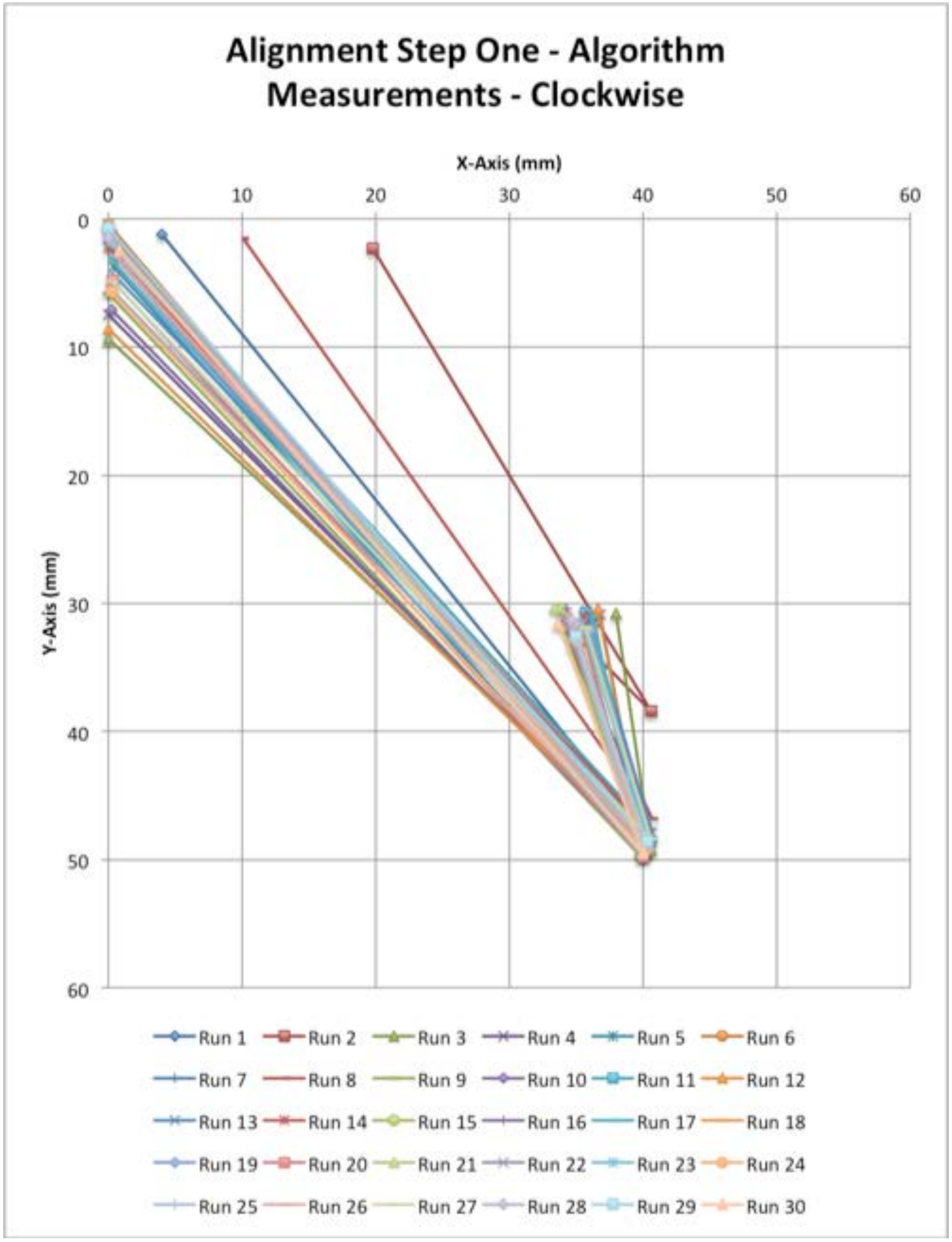


Figure 7.24 Step 1 Algorithm Alignment Measurements - Clockwise Rotation

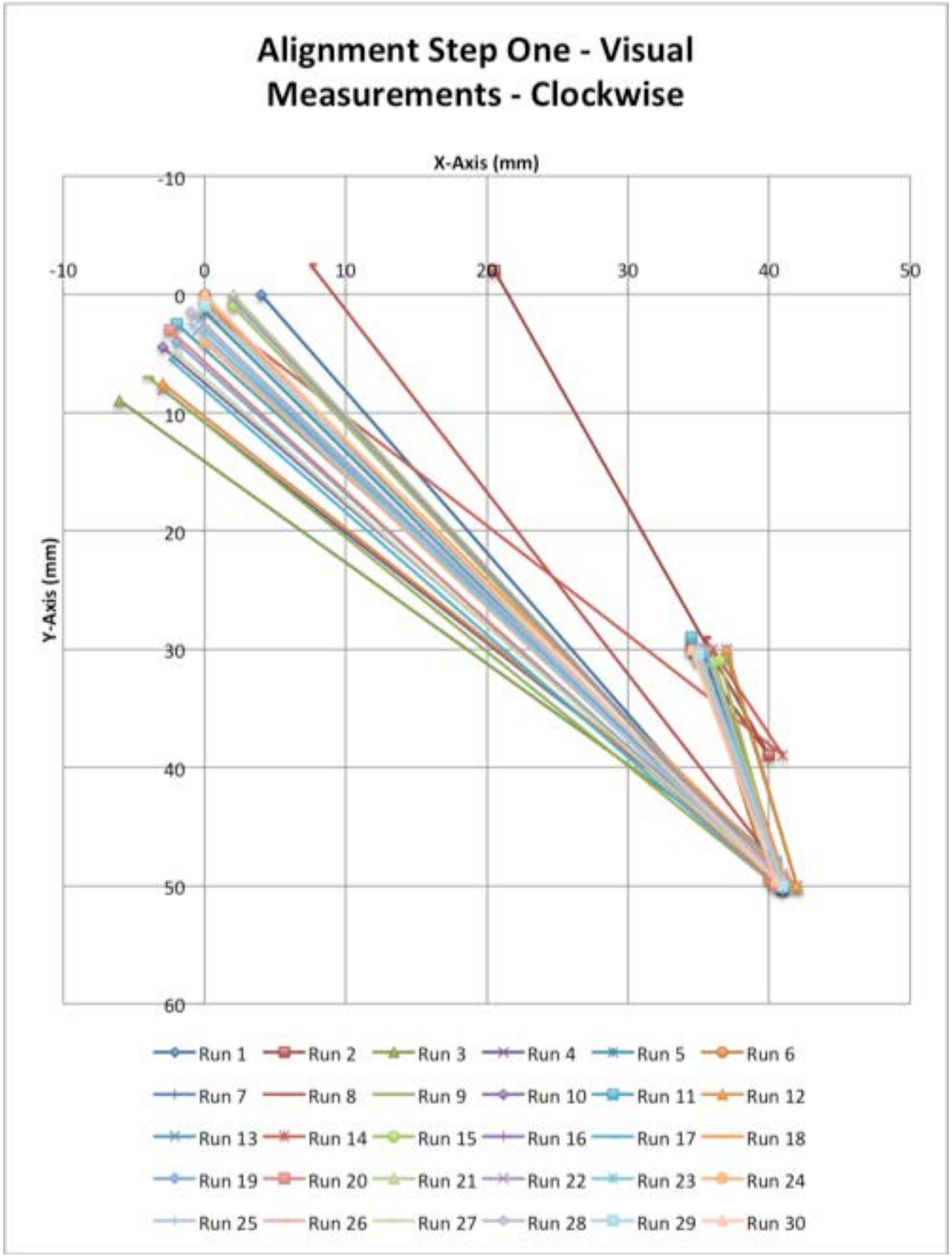


Figure 7.25 Step 1 Visual Alignment Measurements - Clockwise Rotation

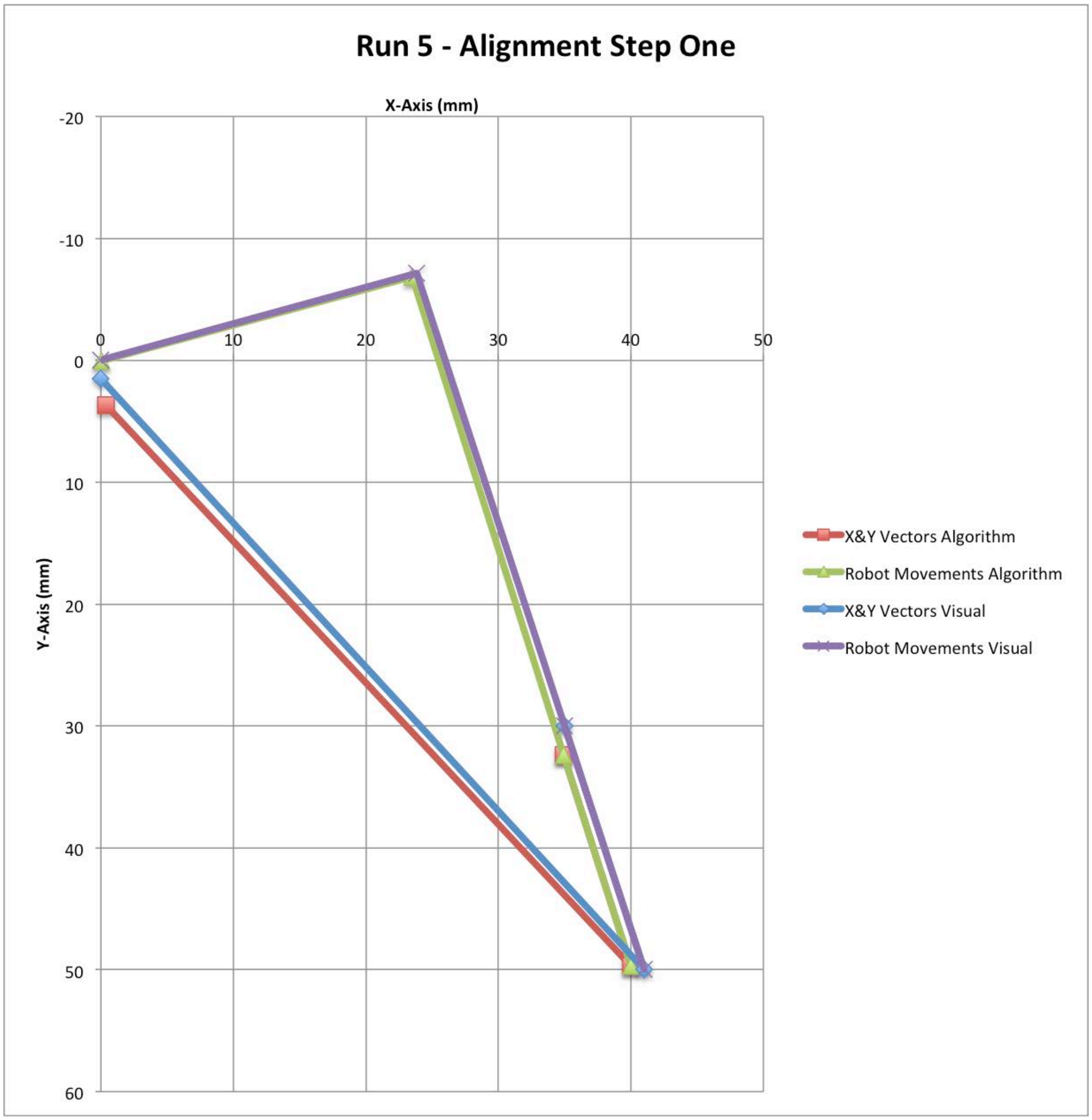


Figure 7.26 Run 5 Step Alignment - Clockwise

7.2.4 Alignment Step 2 - Clockwise

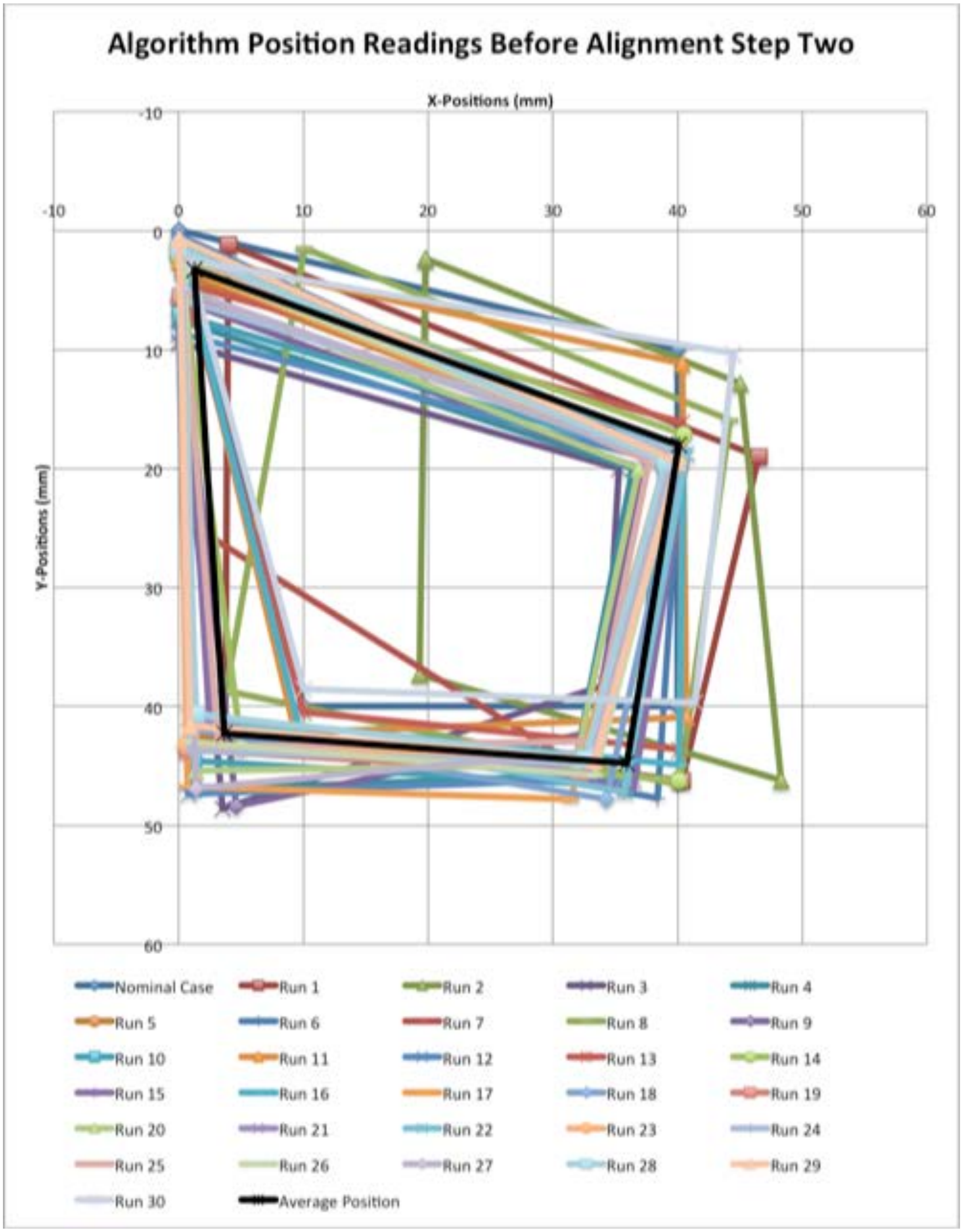


Figure 7.27 Algorithm Reading Laser Positions After Step 1 - 30 Runs - Clockwise

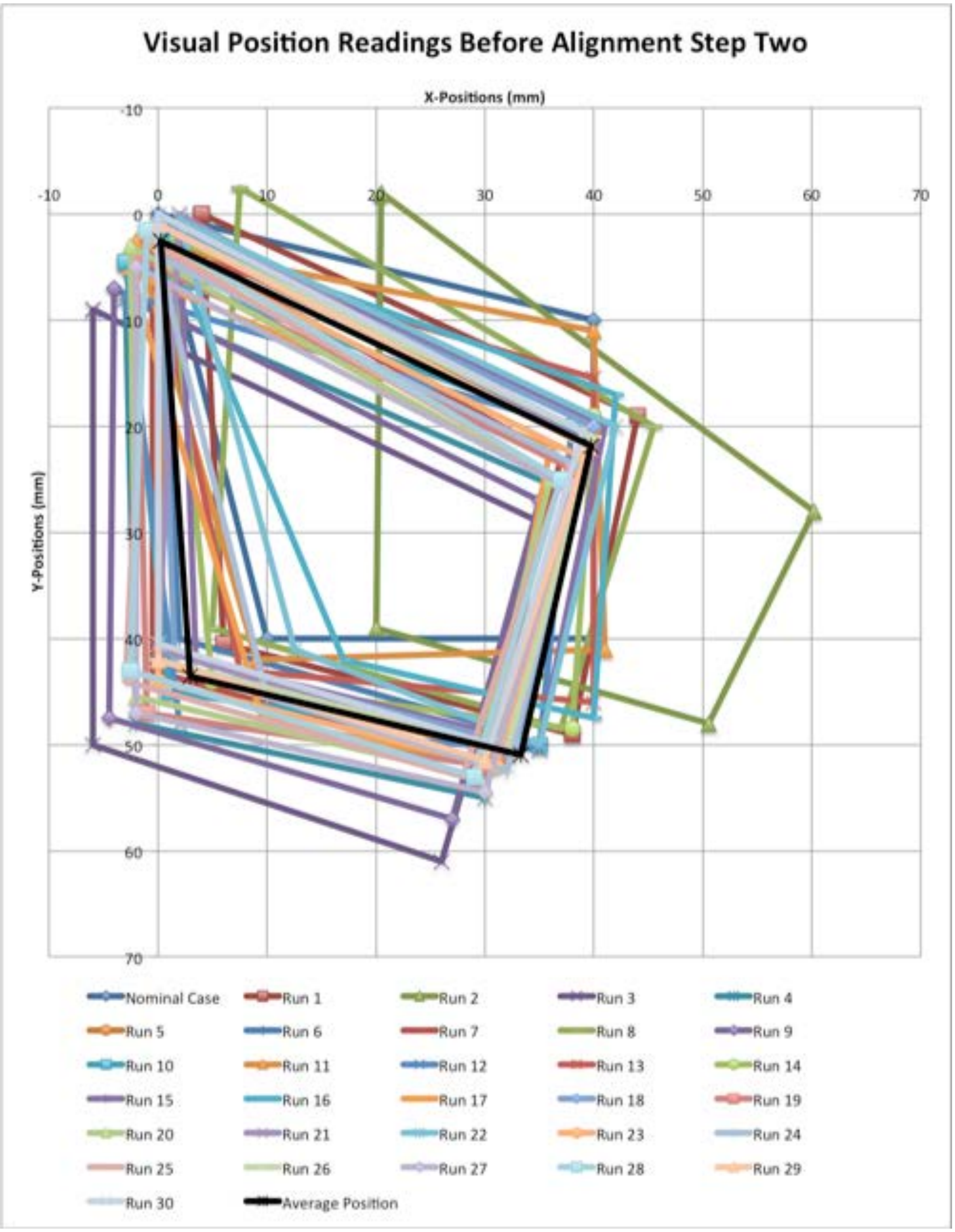


Figure 7.28 Visual Reading Laser Positions After Step 1 - 30 Runs - Clockwise

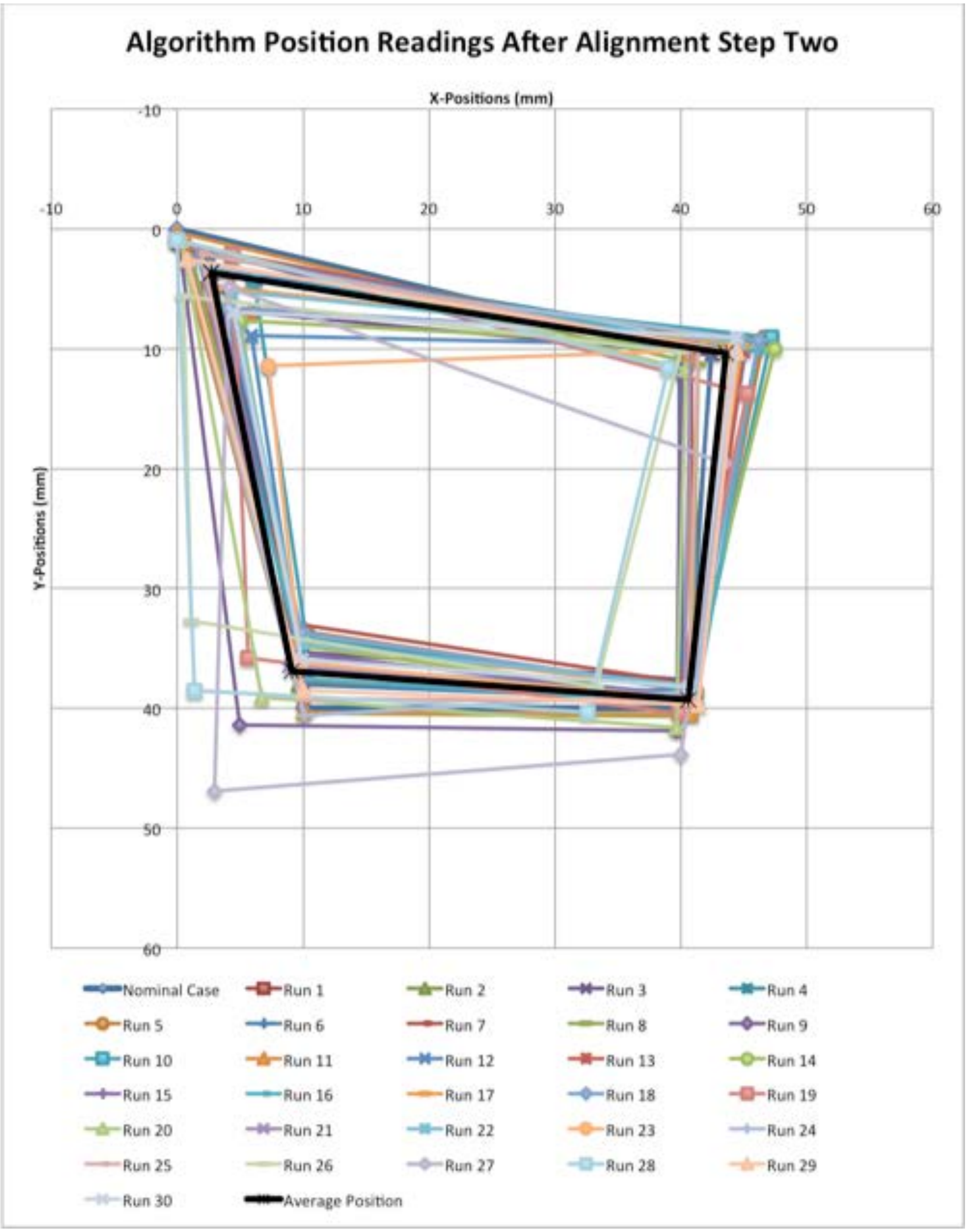


Figure 7.29 Algorithm Reading Laser Positions After Step 2 - 30 Runs - Clockwise

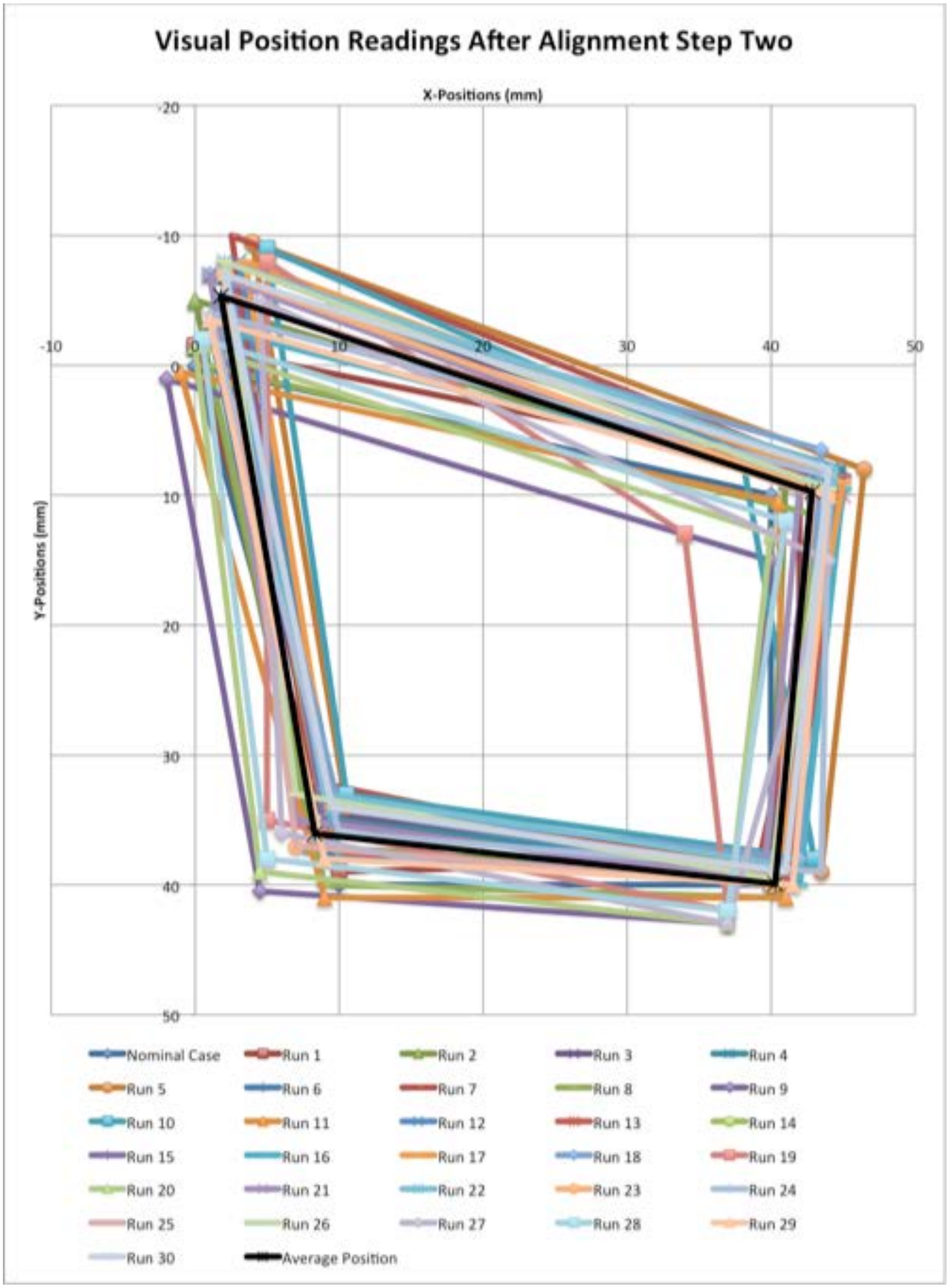


Figure 7.30 Visual Reading Laser Positions After Step 2 - 30 Runs - Clockwise

Figures 7.27 - 7.30 above show the positions of the four lasers as measured by the algorithm and measured visually for both post-Step 1 and post-Step 2 cases of the alignment. As expected, the algorithm locations are all positive, while the visual locations show that the lead laser did go into the negative region. In fact, in a majority of the cases before and after alignment Step 2, the lead laser was in either the negative X or negative Y-region. Before alignment Step 2 the laser was primarily in the negative X-region and the positive Y-region, while after alignment Step 2 a majority of the runs ended with the lead laser in the positive X-region and negative Y-region.

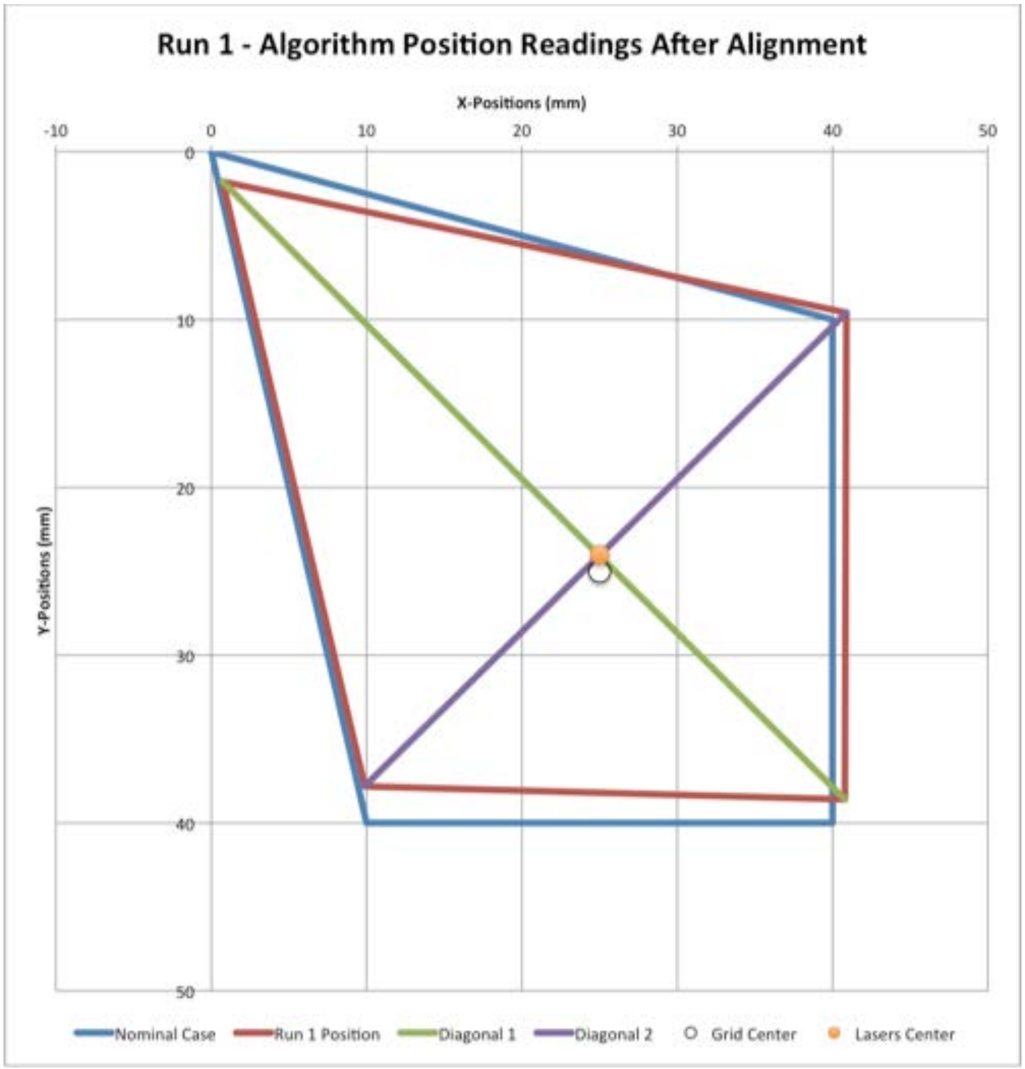


Figure 7.31 Run 1 Aligned Plate Center Position - Algorithm Reading - Clockwise

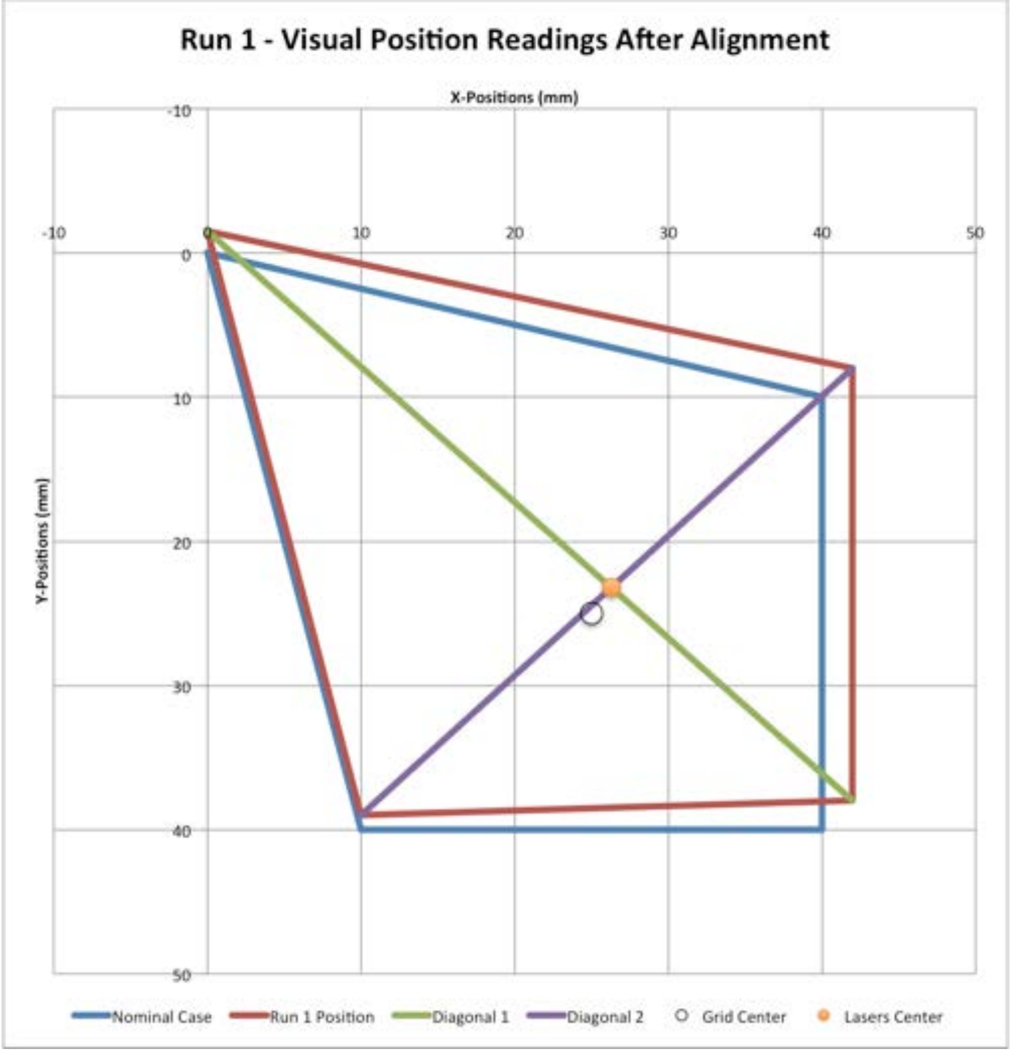


Figure 7.32 Run 1 Aligned Plate Center Position - Visual Reading - Clockwise

The dots in Figure 7.31 shows a final approximate X-error for Run 1 of 0mm and Y-error of 1mm. The two dots in Figure 7.32 show an X-error of approximately 1.3mm and a Y-error of approximately 1.8mm.

The maximum absolute error after Step 1 alignment is 5.3mm in the X-direction with a maximum average error of 1.6mm and a maximum absolute error of 8.2mm in the Y-direction with a maximum average error of 5.3mm. After alignment Step 2 the maximum absolute error is 1.4mm in the X-direction with an average absolute error of 0.7mm and a maximum absolute error of 1.5mm in the Y-direction also with an average absolute error of 0.6mm. Plots of the average error after both Step 1 and Step 2 are shown in Figures 7.33 and 7.34 respectively.

Figures 7.33 and 7.34 show similar error magnitudes to those of Figure 7.20 and 7.21. The average error value after Step 1 is larger than the average error value after Step 2, and the error values are more evenly distributed around the home after Step 2. The similarity in the plots can be seen in Figures 7.35 and 7.36, which show the Counterclockwise and Clockwise Alignment Error plots side-by-side. As required by the error bound of the feedback loop, none of the error in Figure 7.36 exceeds ±1.5mm in the X or Y-directions.

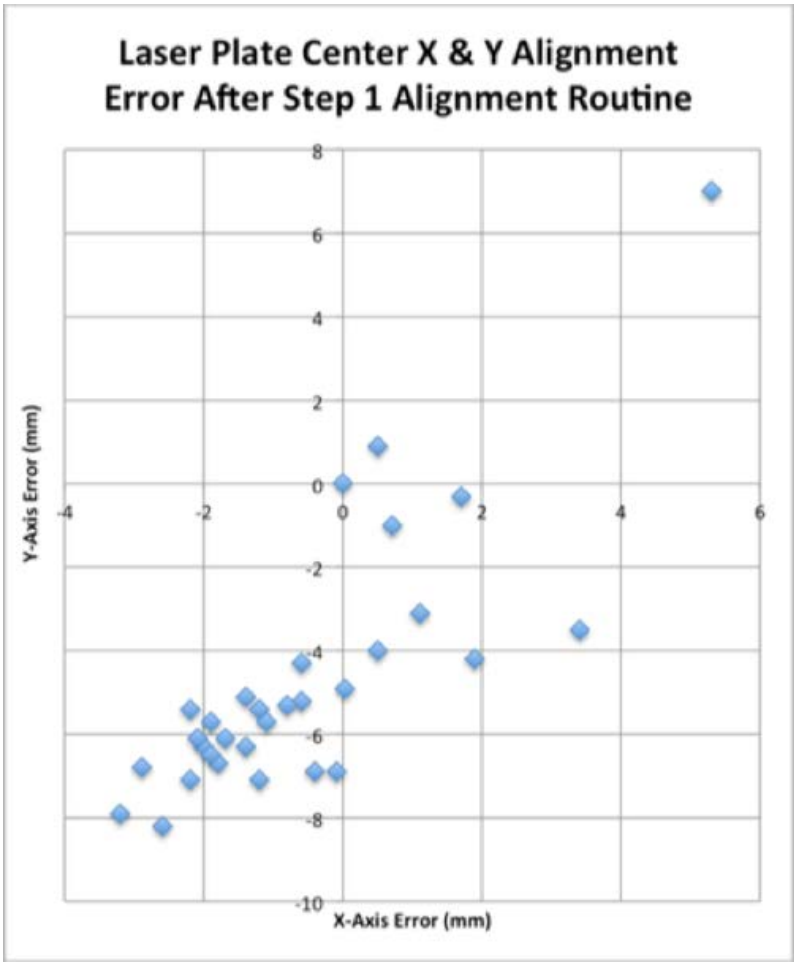


Figure 7.33 Center Point Alignment Error After Step 1 - Clockwise

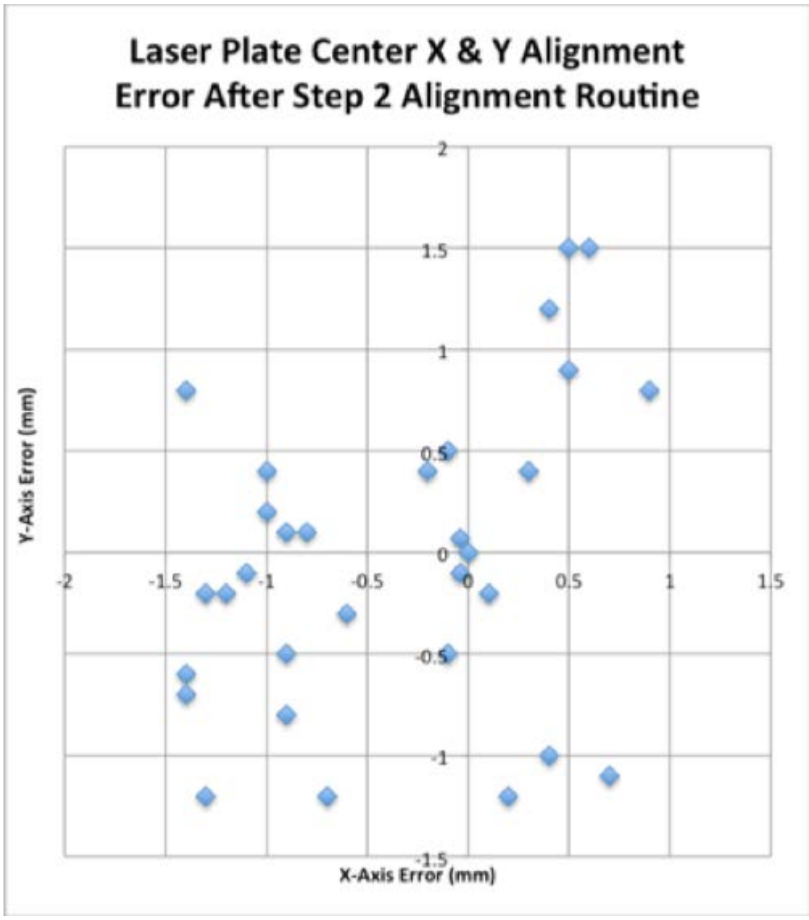


Figure 7.34 Center Point Alignment Error After Step 2 - Clockwise

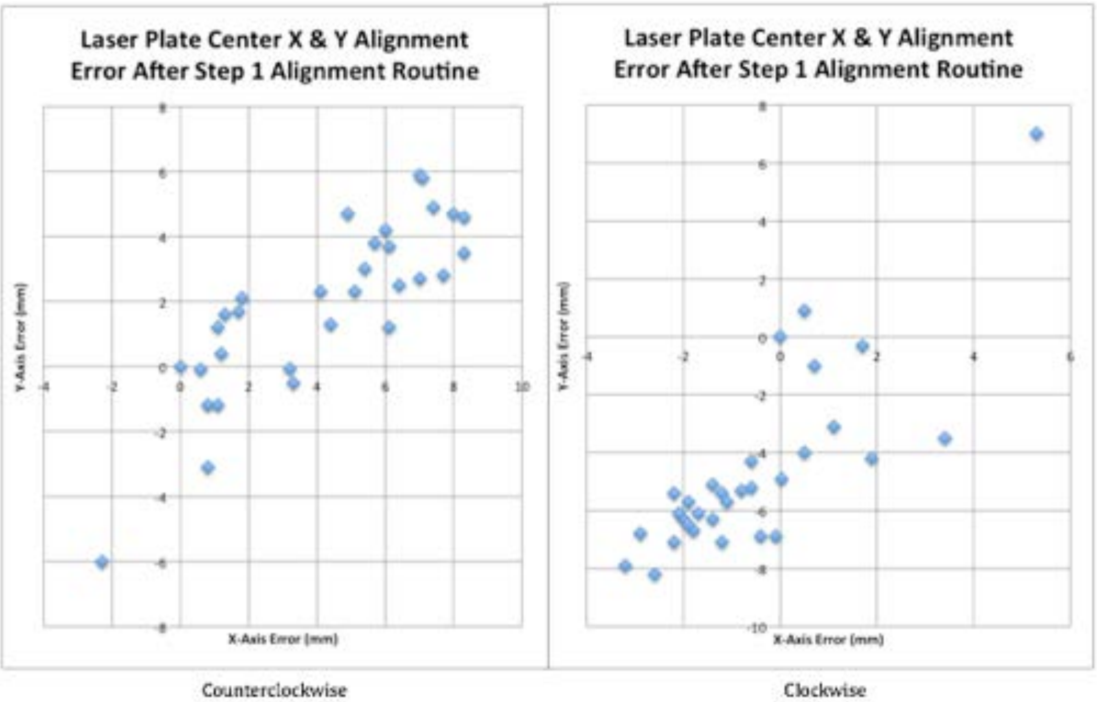


Figure 7.35 Center Point Alignment Error After Step 1 - Counterclockwise vs. Clockwise

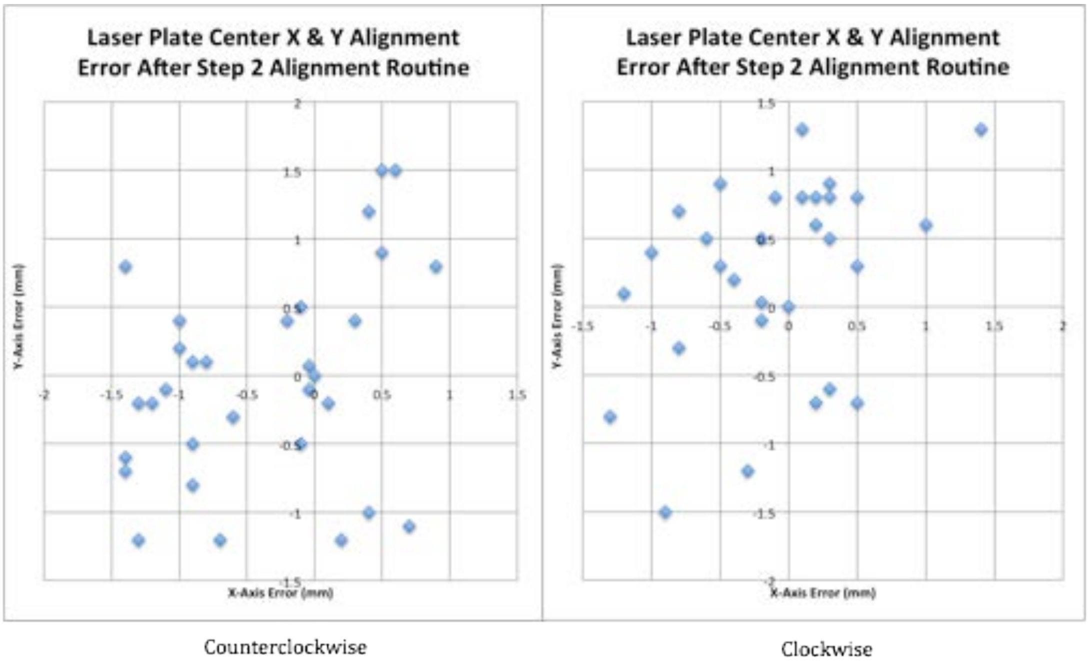


Figure 7.36 Center Point Alignment Error After Step 2 - Counterclockwise vs. Clockwise

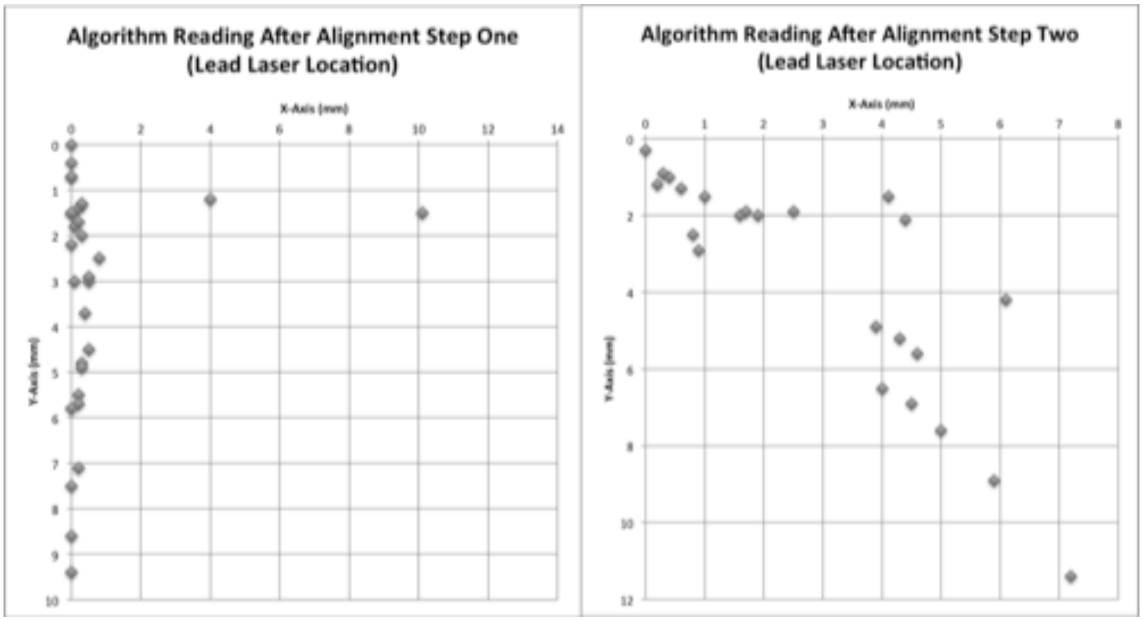


Figure 7.37 Lead Laser Position Before and After Step 2 Alignment - Algorithm Reading - Clockwise

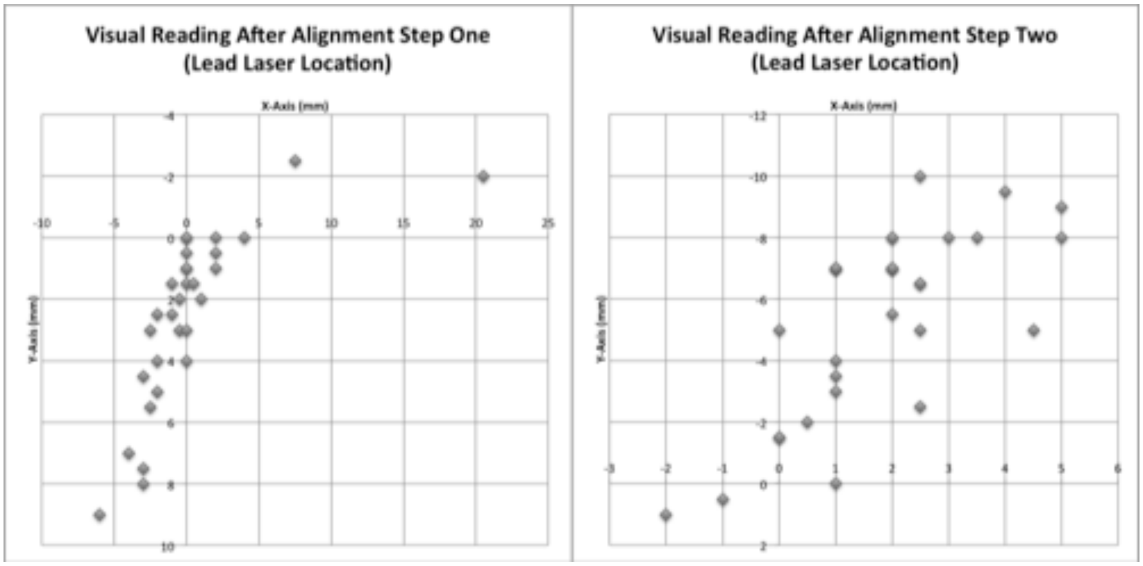


Figure 7.38 Lead Laser Position Before and After Step 2 Alignment - Visual Reading - Clockwise

Five further runs were performed to test the limits of the system. The error bound for Step 2 of the alignment was set to 0.5mm. The system was able to complete the alignment with a maximum X-error of 0.2mm and Y-error of 0.5mm, but took two to three times longer than a 1.5mm error bound run because the feedback loop required additional iterations to achieve a smaller error value. The Algorithm readings and Visual readings of the four lasers at the end of five runs are shown in Figures 7.39 and 7.40 below.

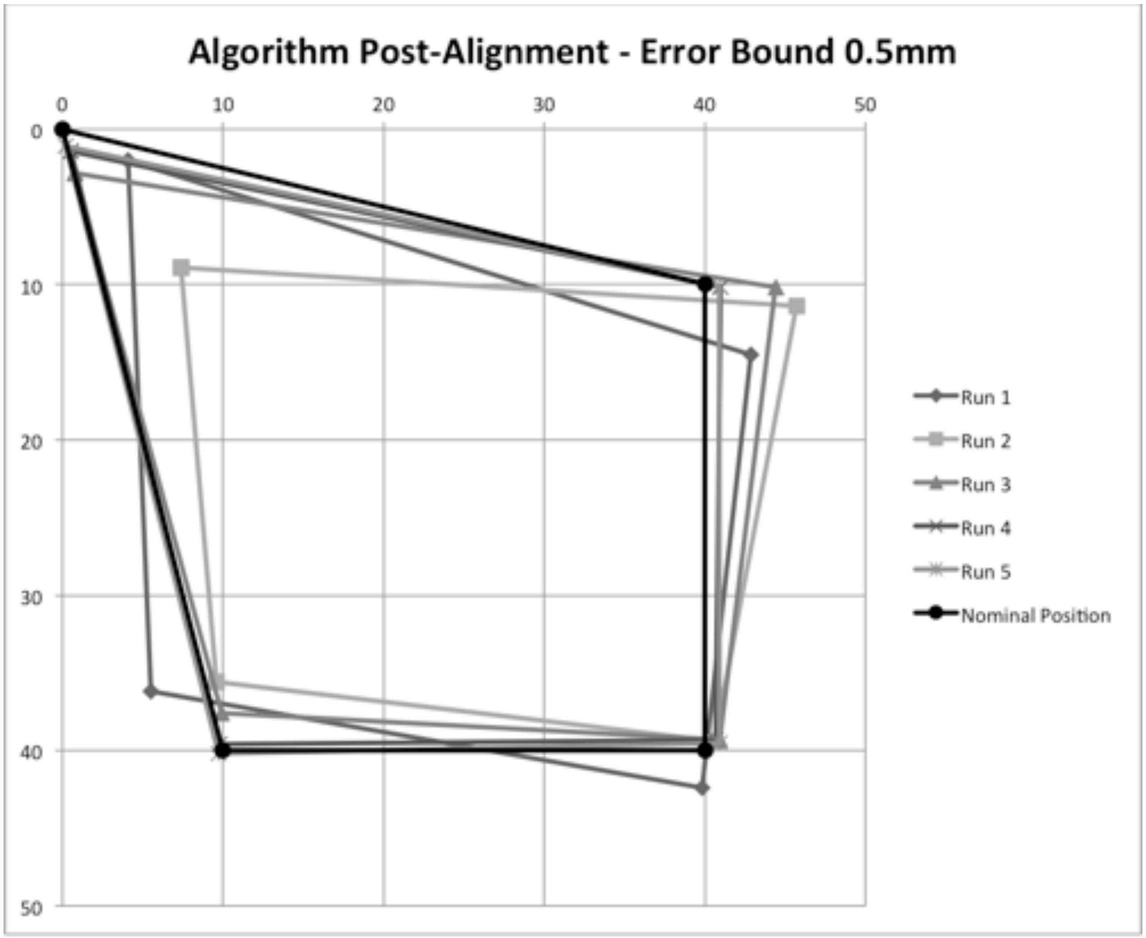


Figure 7.39 Algorithm Reading Laser Positions After Step 2- 5 Counterclockwise Runs with 0.5mm Error Bound

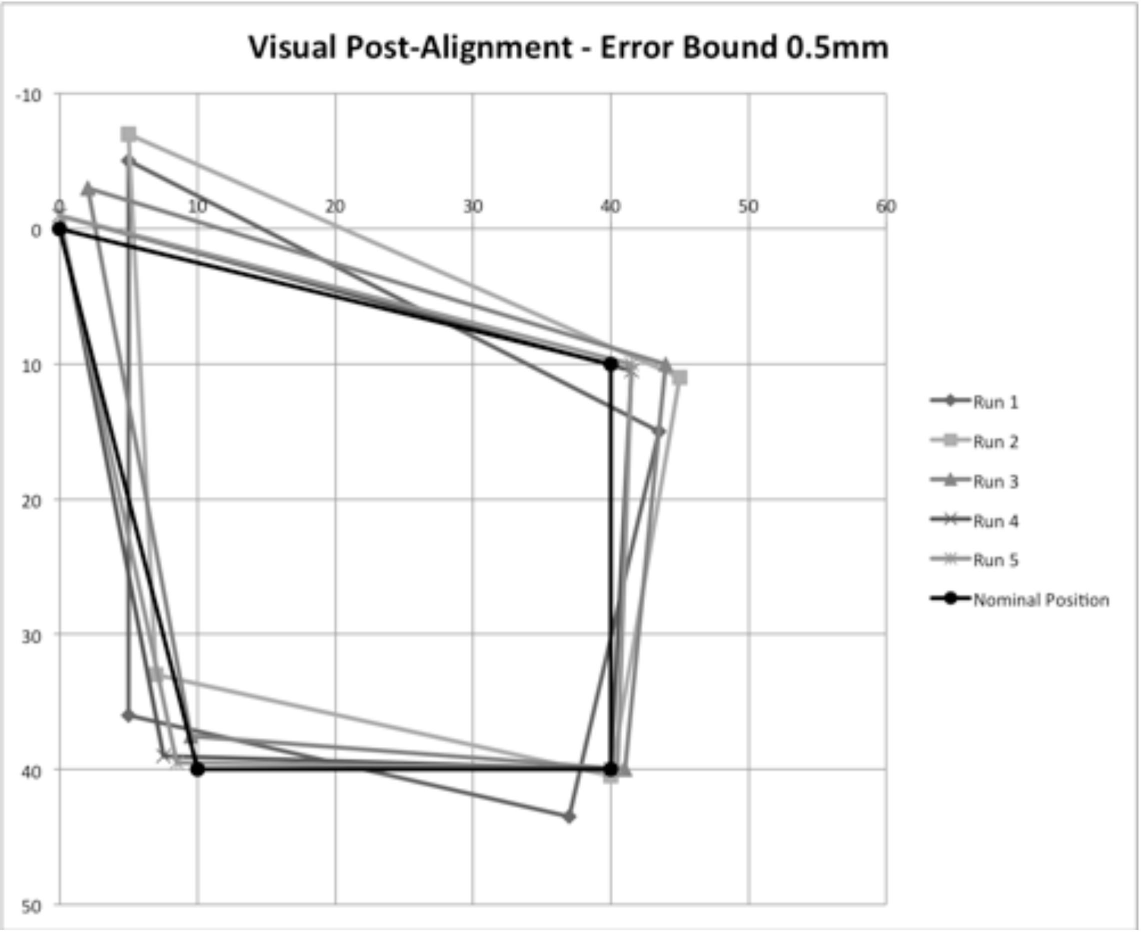
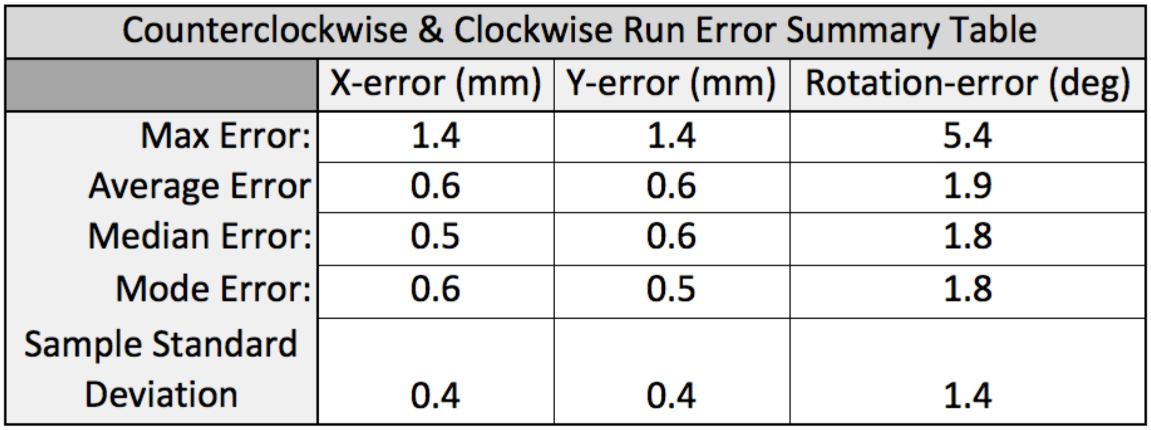


Figure 7.40 Visual Reading Laser Positions After Step 2 - 5 Counterclockwise Runs with 0.5mm Error Bound

The resulting position of each run appears consistent, if not slightly better than those shown above in Figures 7.15, 7.16, 7.29, and 7.30. Additionally, the error bound was lowered to 0.2mm and the system did not converge in a reasonable time. Finally, an error bound of 0.5mm is possible with the current system, but requires additional time. For this reason, an error bound of 1.5mm is more reasonable.

Table 7.2 Combined Counterclockwise and Clockwise Maximum, Average, Median, Mode, Variance, and Sample Standard Deviation of Error for Final Position



Counterclockwise & Clockwise Run Error Summary Table			
	X-error (mm)	Y-error (mm)	Rotation-error (deg)
Max Error:	1.4	1.4	5.4
Average Error	0.6	0.6	1.9
Median Error:	0.5	0.6	1.8
Mode Error:	0.6	0.5	1.8
Sample Standard Deviation	0.4	0.4	1.4

Table 7.2 shows the combined maximum, average, median, mode, and sample standard deviation values of the 30 counterclockwise runs and 30 clockwise runs at the end of Step 2 of each alignment test. The maximum error in both the X and Y-directions is 1.4mm. The table also shows a small average error of 0.6mm in both X and Y-directions with an X-direction standard deviation of 0.6mm and Y-direction standard deviation of 0.4mm.

The resulting center positions values shown in Figure 7.41 demonstrate that all of the alignment runs were within the ±1.5mm error range based on the algorithm readings. The outlined square shown below has a 3mm side lengths centered on the center of the Grid Array representing the error allowance of the alignment system. The position of each of the individual lasers is not as consistent, but this is due to the error in position detection of each of the four lasers using the four quadrants at once. Additionally, the center point is calculated using the intersection of two diagonal lines between the four laser point locations. Therefore, this center point can remain in a smaller area despite a wider variation of the position of each individual laser.

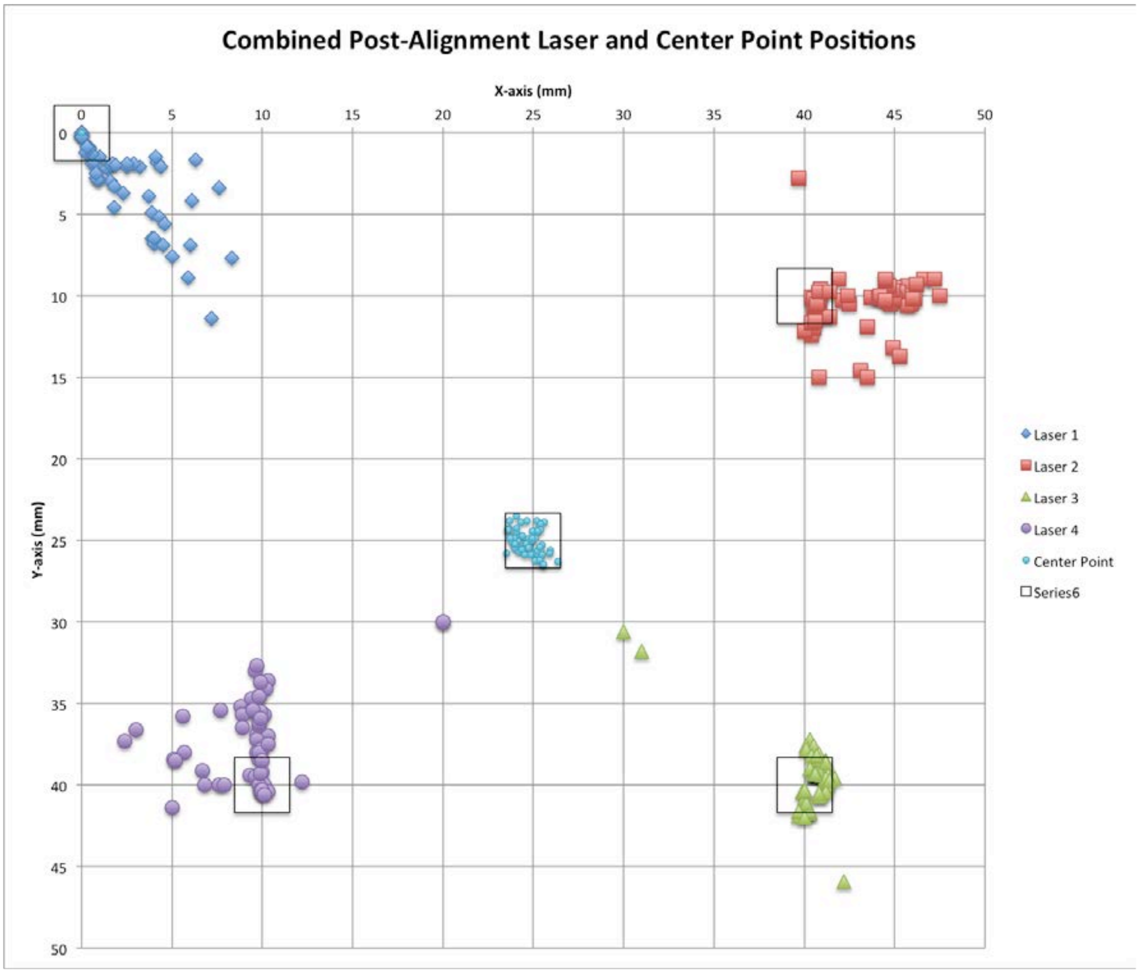


Figure 7.41 Combined Position Results After Alignment for Both Counterclockwise and Clockwise Cases

8. CONCLUSION AND FUTURE RESEARCH

In this thesis a robot alignment system was developed utilizing lasers and a grid-based sensory system. Each alignment step develops a movement path for the robot based on positions detected by the sensory grid. The second step contains a feedback loop that detects the position, moves the robot, and then repeats the process until the position of the robot is within a specified error tolerance.

The system is useful in aligning a robot manipulator, specifically in a pick-and-place automation system. This particular system is designed with the intent of enhancing the alignment procedure of a laboratory diagnostics integrated automation system utilizing a SCARA robot. The robot is used to move titer liquid handling plates from one location to another to allow various diagnostic operations to take place. The alignment system is used in the initial setup of the automation system to teach the various "pick-up" and "drop-off" locations to the robot.

The components of the alignment system include a module held in the robot gripper containing four laser diodes, a module resting in the alignment location containing photodiodes, and a computer-based controller that takes in position data from the photodiodes and outputs movement commands to the robot.

Testing the system by running it through the complete alignment process multiple times shows good results. The sensors can detect the position within 0.5mm of its actual position 74% of the time with a maximum error of 2.9mm in the X-direction and 2.0mm in the Y-direction. The final recursive step of the alignment has an error bound of 1.5mm in the X and Y-directions. It has an overall final absolute average positional X-error of

0.6mm and Y-error of 0.6mm with a maximum error of 1.4mm for both the X and Y-directions. A small number of additional full alignment tests were run with a final error bound of 0.5mm in the X and Y-directions. The system achieved this with a maximum X-error of 0.2mm and Y-error of 0.5mm, but took longer to do so than in the 1.5mm case, showing that a high accuracy of the overall alignment is possible, but more time consuming. By comparing the average error value from the 1.5mm error allowance case to the average error value of the 0.5mm error allowance case, it is not surprising that the system is capable of achieving a 0.5mm error since the average error with a 1.5mm error allowance is 0.6mm.

Although the system performed better than expected for a grid spacing of 10mm from sensor center to sensor center, it is postulated that its performance would be further enhanced with a higher density sensor grid. Furthermore, there were small misalignments in the positioning of the lasers. Therefore, finding a method of aligning them would improve the alignment accuracy of the system. This could include screw adjustments that would chance the tilt and the position of the lasers.

Currently the system only functions in a low lighting setting. In this settings a software filter can be set to a low enough value to remove the effect of the ambient light while retaining the ability to accurately calculate position. Nonetheless, having a system that would work in any lighting condition would make the system more practical. A future improvement is to add either a light filter over the photodiodes to restrict the light bandwidth striking them, or to select photodiodes with a narrow light detection bandwidth and use corresponding laser diodes.

Additionally, the system would benefit from an updated position detection algorithm for detecting up to four laser points. The current method is prone to error in the boundary between the quadrants of the grid. If one laser point is close to a boundary, it will likely be detected by two quadrants adding substantial error to the positioning. Currently this is compensated for by operating the system in configurations that will decrease the possibility of a laser shining into two quadrants, and by adding a feedback loop to adjust the alignment until it is within the error bound.

A similar problem can occur in the first alignment step. The system should detect only the lead laser on the edge of the Grid Array. If one or two of the other three lasers is close to the edge of the grid, it will potentially shine on the Grid Array, altering the measured position. To eliminate this, a future update would include software control of the lasers that would turn off the three unused lasers during the first alignment step and then turn them back on to complete the alignment.

Expanding the system to detect and align in 3-dimensions would enhance the functionality of the system. One such method of doing this is by using hall-type sensors to measure small Z-axis movements needed to position the plate. This additional functionality would allow the system to perform more of the alignment for teaching a location. Currently the Z-axis alignment must be performed manually. With this new method, the laser plate could be set above the Grid Array and the system would align in the X, Y, and Z-directions.

LIST OF REFERENCES

LIST OF REFERENCES

[1] Newport Corporation. "*Laser Diode Technology,*" Newport, 2013. [Online]. Available: http://www.newport.com/Tutorial-Laser-Diode-Technology/852182/1033/content.aspx [Accessed: Nov. 4, 2013].

[2] OSI Optoelectronics. "*Photodiode Characteristics and Applications. (OSI Optoelectronics),*" OSI Optoelectronics, 2013. [Online]. Available: http://www.osioptoelectronics.com/application-notes/AN-Photodiode-Parameters-Characteristics.pdf [Accessed: Nov. 3, 2013]

[3] Oxford University Press. "*Definition of collimate in English,*" Oxford Dictionaries, 2013. [Online]. Available: http://www.oxforddictionaries.com/definition/english/collimate [Accessed: Nov. 4, 2013].

[4] Oxford University Press. "*Definition of concatenate in English,*" Oxford Dictionaries, 2013. [Online]. Available: http://www.oxforddictionaries.com/definition/english/concatenate?q=concatenate [Accessed: Nov. 4, 2013].

[5] M. W. Spong and S. Hutchinson, *Robot Modeling and Control*. United Stats of America: John Wiley & Sons, Inc., 2006.

[6] R. Singh, R. M. Voyles, D. Littau, and N. P. Papanikolopoulos, "Pose Alignment Of An Eye-In-Hand System Using Image Morphing" in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference*, Vol. II, 1998, pp. 698-705.

[7] R. Horaud, F. Dornaika, and B. Espiau, "Visually Guided Object Grasping," *(M. F. INRIA, Ed.) Robotics and Automation, IEEE Transactions*, vol. 14 (4), pp. 525-532

[8] G. H. Beckhart, P. R. Conarro, K. J. Harrell, M. C. Krause, and K. M Farivar-Sadri, "*Robot Alignment System and Method,*" U.S. Patent 6 532 403 B2, March 11, 2003.

[9] HighRes Biosolutions. "*MicroTeach,*" highresbio.com, 2010. [Online]. Available: www.highresbio.com [Accessed: July 14, 2012].

[10] Y. Kawato and W. Kim, "A Novel Multi-DOF Precision Positioning Methodology Using Two-axis Hall-effect Sensors," in American Control Conference, 2005, Vol. V, 2005, pp. 3042-3047.

[11] "Position Sensing and Alignment Engineering Guide," Duma optronics Ltd., Nesher, Isreal.

[12] N.P. Papanikolopoulos, "The Framework of Controlled Active Vision," Mathematical Computational Modeling, vol. 24, pp. 145-163, 1996.

[13] S. C. Jacobsen, I. D. McCammon, K. B. Biggers, and R. P Phillips, "Tactile Sensing System Design Issues in Machine Manipulation," in *Robotics and Automation, Proceedings. 1987 IEEE International Conference*, Vol. IV, 1987, pp. 2087-2096.

[14] S. Ohara, Y. Yanagihara, and T. Takahashi, "Robot Operation Based on a Model Updated by Sensory Data," in *Advanced Robotics, 1991. 'Robots in Unstructured Environments' 91 ICAR., Fifth International Conference*, Vol. I, 1991, pp. 1658-1661.

[15] "Setup Guide & User's Manual Revision 2.2 PreciseFlex 400 SCARA on SAMI EX System," Beckman Coulter, Inc. Brea, CA, Revision 2.2, Aug. 22, 2013.

[16] Berkeley Lab, A U.S. Department of Energy National Laboratory Operated by the University of California. "Laser Classification Explanation," lbl.gov, Aug. 2, 2013. [Online]. Available: http://www.lbl.gov/ehs/safety/lasers/classification.shtml [Accessed: Oct. 31, 2013].

[17] Rockwell Laser Industries, Inc. "*Laser Standards and Classification,*" Rockwell Laser Industries, 2013. [Online]. Available: http://www.rli.com/resources/articles/classification.aspx [Accessed: OCt. 31, 2013].

[18] Arduino. "Arduino Mega 2560," Arduino.cc, 2013. [Online]. Retrieved 2012, from Arduino: http://arduino.cc/en/Main/ArduinoBoardMega2560 [Accessed: Nov. 4, 2013].

[19] Peak Robotics, Inc. "*KiNEDx Laboratory Robot: User's Manual*," paa, Peak Analysis & Automation, Jan. 8, 2008. [Online]. Available: http://www.paa-automation.com/labauto/information/p-download-application-notes.asp , File: KiNEDx KX-Series User's Manual, RevL.pdf [Accessed: Dec. 2012].

[20] Roithner Lasertechnik GmbH. “*Diverse Laser Diodes,*” Roithner Lasertchnik, Dec. 20, 2011. [Online]. Available: http://www.roithner-laser.com/datasheets/ld_div/adl-65077su.pdf [Accessed: June 2012]

[21] Roithner Lasertechnik GmbH. “*Optics for Laser Diodes,*” Roithner Lasertchnik. [Online]. Available: http://www.roithner-laser.com/datasheets/ld_optic/LC5.pdf [Accessed: June 2012]

[22] Texas Instruments Inc. “*Monolithic Photodiode and Single-Supply Transimpedance Amplifier,*” Texas Instruments, Jan. 1994. [Online]. Available: www.ti.com/lit/ds/symlink/opt101.pdf [Accessed: Spt. 2012]

[23] Arima Lasers. “Laser Components,” lasercomponents.com, 2005. [Online]. Available: http://www.lasercomponents.com/de/?embedded=1&file=fileadmin/user_upload/home/Datasheets/divers-vis/ari/655nm/adl-65104tl.pdf&no_cache=1 [Accessed: Oct. 18, 2013].

[24] S. M. Ross, *Introduction to Probability and Statistics for Engineers and Scientists*. 4th ed. Burlington, MA: Elsevier Academic Press, 2009

[25] Peak Robotics, Inc. “*KiNEDx KX-300 Laboratory Robot: Software Instructions*,” paa, Peak Analysis & Automation, June 3, 2009. [Online]. Available: http://www.paa-automation.com/labauto/information/p-download-application-notes.asp , File: KiNEDx Robot Control Software Instructions v3.0.15 .NET.pdf [Accessed: Dec. 2012].

[26] Malzahn, U. M. (2008, November 11). “*Driving laser diodes: Comparing discrete vs. integrated circuits,*” iC-Haus, EE Times [Online]. Available: http://www.eetimes.com/document.asp?doc_id=1274119 [Accessed: Nov. 4, 2013].

APPENDICES

Appendix A Specific Robot Integration Using the Peak KiNEDx Robot with LabVIEW

The first function used to control the robot is the Initialize function shown in Figure A.1. It sets up the communication between the robot and the computer by opening the appropriate COM port. The function receives two inputs that are consistent for every function. The first input is a command line that sends the run command to each function. The next input is an error line that allows any errors messages or status indications to be passed to each function. The following line output the status of the function to the GUI. The next input reads in the file name of the robot parameters to make them available for use in controlling the robot. The next input is a Boolean input for loading motor k parameters. The following input is the file path to the teach point file and is followed by an input that receives a *concatenated* string listing the order that the axes should be initialized in. Following this are three more Boolean inputs. The first determines if the motors should be homed upon initialization, the second determines if the parameter file should be reset, and the third determines if the motors should be re-homed if they are already homed. The final input is the sequence file name to allow the system to locate the sequence file. Finally, the function outputs any necessary error messages.

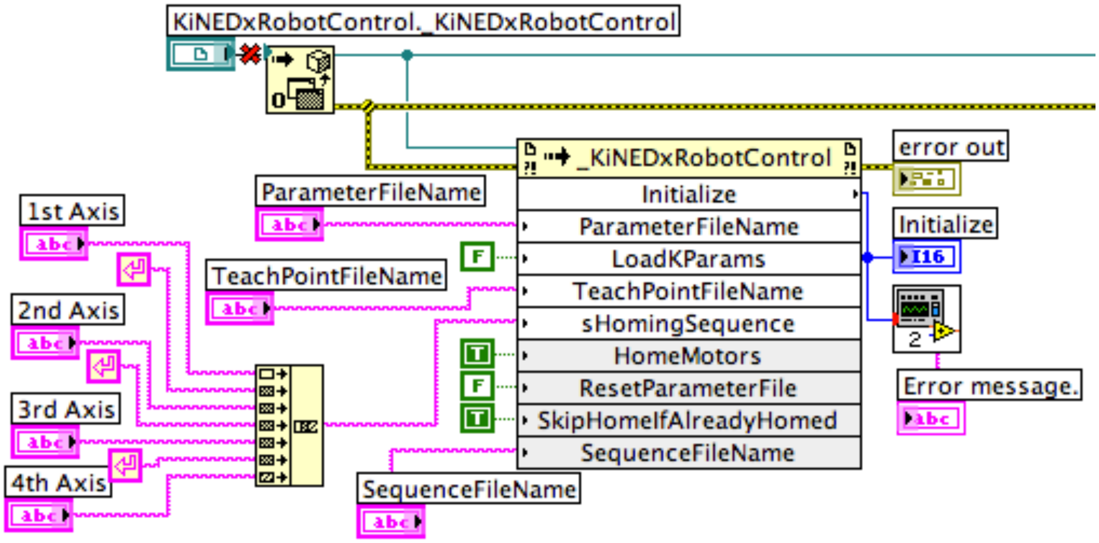


Figure A.1 Robot Initialization Function

The parameter file provides the robot with its travel limits along each axis. It can also be used to revers the home position of an axis from one limit end to the other. The teach point file is used to store values of positions taught to the robot, and finally the sequence file stores the motion files that the robot follows when running.

Multiple parameters, like the homing order shown above in Figure A.1, are passed into a function block by *concatenating* the values into one input of data type string. Figure A.2 shows an example of four values being concatenated into one input with carriage return characters between them to divide up the values and then being passed into a single input.

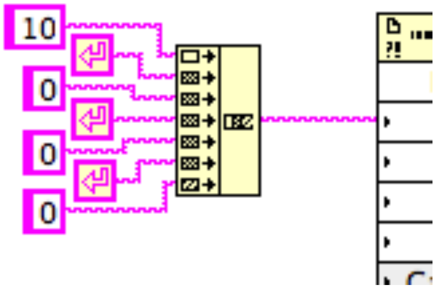


Figure A.2 Concatenation Example

The MoveRelativeCartesian function shown in Figure A.3 below requires eleven inputs. It takes in the two standard inputs. Then it requires a string type input that passes a concatenated string that contains any X, Y, Z, and theta movement values for the robot to execute. The movement velocity and acceleration are then input. Additionally, it reads in a Boolean command that decides if the system should wait until this function finishes running to start the next function. Then it receives a value for the calculated move time, and then another Boolean value deciding if the function should notify the system when the movements have been performed. It also takes in a value that notifies the system which coordinate system to use. The final two inputs deal with reference frame modifications. The first of these applies a rotation to the reference frame being used. The last input is the offset used for the center of the robot tool when using tool reference frame. The function then executes by moving the robot and displays any necessary error messages. It can also display an additional output that lists the length of time that it took to run the function.

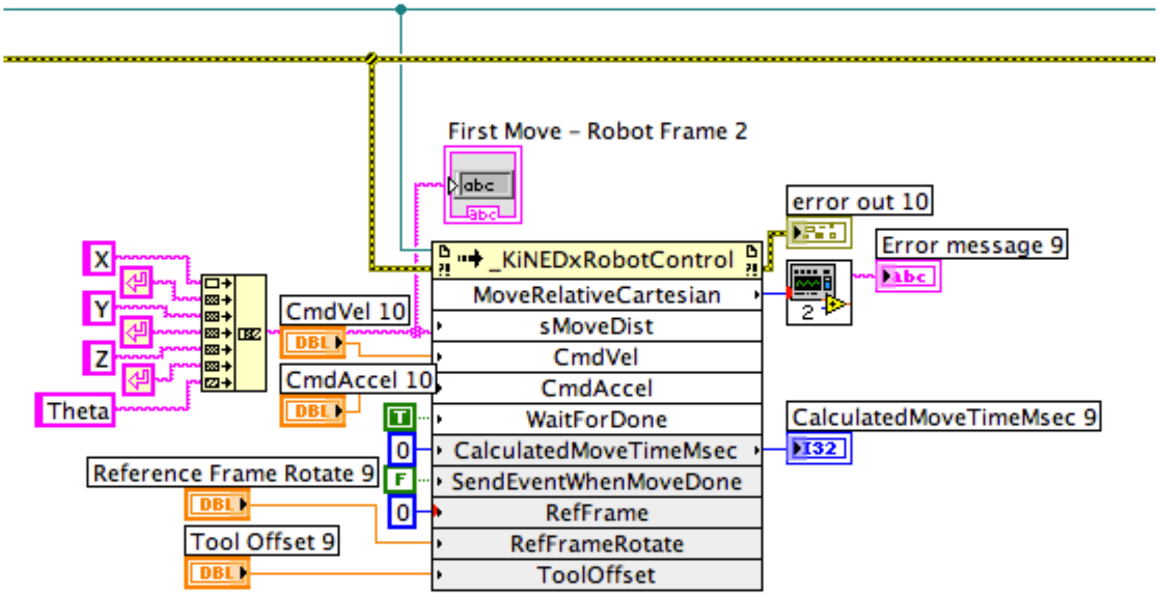


Figure A.3 Robot MoveRelativeCartesian Function

The three reference frame options above are the world reference frame specified by the value 0 which is the coordinate system based on the robot base, the tool reference frame specified by 1 which is related to the robot tool or gripper, and the rotated world reference frame specified by 2 which is the same as the world reference frame but rotated by the value indicated in the RefFrameRotate input.

The robot ShutDown function shown below in Figure A.4 receives the two standard inputs, the control/data wire, and the error out/status wire. The only output is an error out signal. When the function receives the shutdown command, it closes the COM port communication between the robot and computer and then sends a signal to the error out block to report the results of the function.

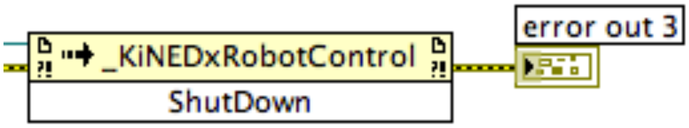


Figure A.4 Robot ShutDown Function

Appendix B Robot System Compatibility

To work with other robotic platforms, functions specific to each robot would to be brought into LabVIEW. The essential commands would be to initialize the connection with the robot, move the robot along the robot frame Cartesian coordinate system, move the robot along the tool frame Cartesian coordinate system, and then close the connection. The system does not require any feedback from the robot, minimizing the integration between LabVIEW and the robot controller.

Appendix C Component Cost and Suppliers

Component Suppliers:

Electrical Circuitry components: Digi-Key Corporation (www.digikey.com)

Photodiodes: Digi-Key Corporation (www.digikey.com)

Laser Diodes: Roithner Lasertechnik GmbH (http://www.roithner-laser.com)

Laser Collimating Lenses: Roithner Lasertechnik GmbH (http://www.roithner-laser.com)

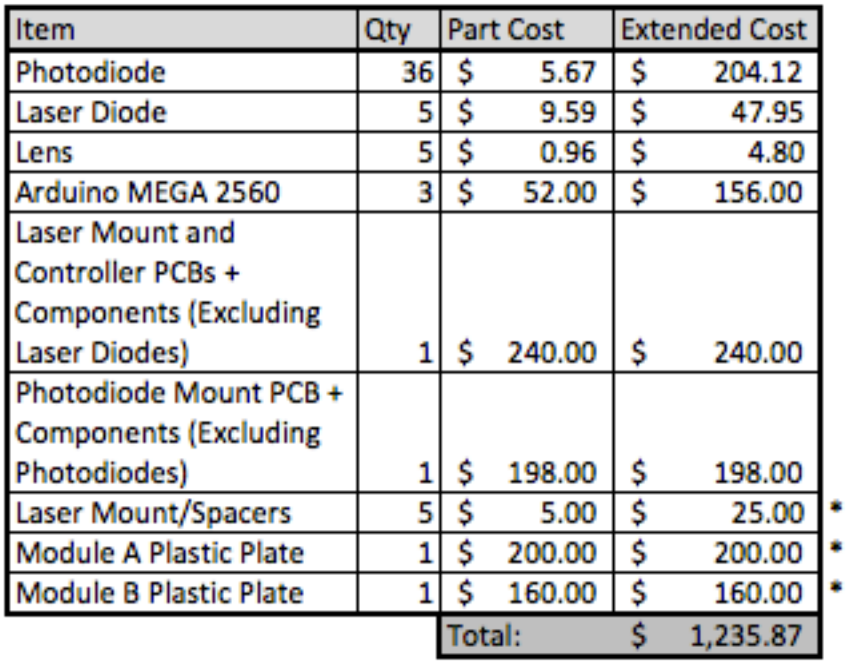
Arduino MEGA 2560: Digi-Key Corporation (www.digikey.com)

Printed Circuit Board for Laser Mounts and Laser Controller: ExpressPCB (http://www.expresspcb.com)

Printed Circuit Board for Photodiode Mounts: ExpressPCB (http://www.expresspcb.com)

Cost Approximation:

Table D.1 System Cost Estimate (high-level components)



Item	Qty	Part Cost	Extended Cost	
Photodiode	36	$ 5.67	$ 204.12	
Laser Diode	5	$ 9.59	$ 47.95	
Lens	5	$ 0.96	$ 4.80	
Arduino MEGA 2560	3	$ 52.00	$ 156.00	
Laser Mount and Controller PCBs + Components (Excluding Laser Diodes)	1	$ 240.00	$ 240.00	
Photodiode Mount PCB + Components (Excluding Photodiodes)	1	$ 198.00	$ 198.00	
Laser Mount/Spacers	5	$ 5.00	$ 25.00	*
Module A Plastic Plate	1	$ 200.00	$ 200.00	*
Module B Plastic Plate	1	$ 160.00	$ 160.00	*
		Total:	$ 1,235.87	

*Component price estimated based on rapid prototyping by a third party (including overhead, material, and labor).

Appendix D Component Specification Reference

Arduino MEGA 2560: [18]

Robot Specifications: [19]

Laser diode specification sheet: [20]

Lens specification sheet: [21]

Photodiode specification sheet: [22]

Appendix E In-Text Notes

1. Titer Plate

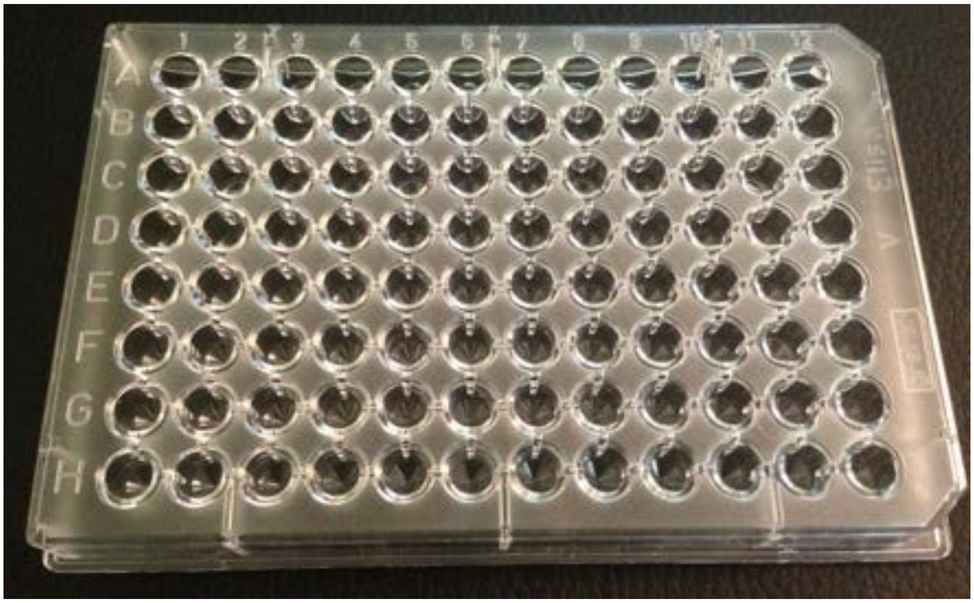


Figure E.1 Titer Plate

2. Pictures Supplied with permission from Beckman Coulter, Life Science Division Headquarters, 5350 Lakeview Parkway S. Drive, Indianapolis, In 46268 (October 25, 2013)

3. Pictures Supplied with permission from Beckman Coulter, Life Science Division Headquarters, 5350 Lakeview Parkway S. Drive, Indianapolis, In 46268 (October 25, 2013)

4. Laser diode picture from Arima Lasers [23]

5. Arduino Board from Arduino Website [18]

6. Robot picture taken from KiNEDx Robot User's manual Pg. 12 [19]

7. Robot picture taken from KiNEDx Robot User's manual Pg. 46 [19]

8. Robot picture taken from KiNEDx Robot User's manual Pg. 47 [19]